Proceedings of the 10th International Workshop on Spoken Language Translation (IWSLT 2013)

Heidelberg, Germany.
Dec. 5-6, 2013

Edited by: Joy Ying Zhang

Table of Contents

Foreword

The International Workshop on Spoken Language Translation (IWSLT) is an annually scientific workshop, associated with an open evaluation campaign on spoken language translation, where both scientific papers and system descriptions are presented. The 10th International Workshop on Spoken Language Translation takes place in Heidelberg, Germany on Dec. 05 and 06, 2013.

The IWSLT includes scientific papers in dedicated technical sessions, either in oral or poster form. The contributions cover theoretical and practical issues in the field of Machine Translation (MT), in general, and Spoken Language Translation (SLT), including Automatic Speech Recognition (ASR), Text-to-Speech Synthesis (TTS) and MT, in particular:

- Speech and text MT
- Integration of ASR and MT
- MT and SLT approaches
- MT and SLT evaluation
- Language resources for MT and SLT
- Open source software for MT and SLT
- Adaptation in MT
- Simultaneous speech translation
- Speech translation of lectures
- Spoken language summarization
- Efficiency in MT
- Stream-based algorithms for MT
- Multilingual ASR and TTS
- Rich transcription of speech for MT
- Translation of on-verbal events

Submitted manuscripts were carefully peer-reviewed by members of the program committee and papers were selected based on their technical merit and relevance to the conference. The large number of submissions as well as the high quality of the submitted papers indicates the interest on Spoken Language Translation as a research field and the growing interest in these technologies and their practical applications.

The results of the spoken language translation evaluation campaigns organized in the framework of the workshop are also an important part of IWSLT. Those evaluations are organized in the manner of competition. While participants compete for achieving the best result in the evaluation, they come together afterwards and discuss and share their techniques that they used in their systems. In this respect, IWSLT proposes challenging research tasks and an open experimental infrastructure for the scientific community working on spoken and written language translation. This year the IWSLT evaluation offered a very challenging and appealing task on spoken language

translation of public speeches in a variety of topics, including a dedicated task to automatic speech recognition in order to cover the full pipeline of speech translation.

For each task, monolingual and bilingual language resources, as needed, are provided to participants in order to train their systems, as well as sets of manual and automatic speech transcripts (with n-best and lattices) and reference translations, allowing researchers working only on written language translation to also participate. Moreover, blind test sets are released and all translation outputs produced by the participants are evaluated using several automatic translation quality metrics. For the primary submissions of all MT and SLT tasks a human evaluation was carried out as well. Each participant in the evaluation campaign has been requested to submit a paper describing his system, the utilized resources. The organizers present a survey of the evaluation campaigns.

Welcome to Heidelberg!

Alex Waibel, General Chair IWSLT 203

Organizers

Workshop Chair

Alex Waibel, KIT&CMU

Joseph Mariani, LIMSI-CNRS & IMMI

Evaluation Chair

Marcello Federico, FBK

Sebastian Stüker, KIT

Program Chair

Joy Zhang, CMU

Publicity Chair

Eiichiro Sumita, NICT

Chiori Hori, NICT

Local Chair

Margit Rödder, KIT

Program Committee

- Alexandre Allauzen (LIMSI, France)
- Loic Barrault (LIUM, France)
- Laurent Besacier (LIG, France)
- Mauro Cettolo (FBK, Italy)
- Boxing Chen (NRC, Canada)
- Chris Dyer (CMU, USA)
- Matthias Eck (Facebook, USA)
- Ge Gan (Qualcomm, USA)

- Xiaodong He (Microsoft Research, USA)
- Fei Huang (IBM, USA)
- Qun Liu (ICT, China)
- Yang Liu (Tsinghua Univ., China)
- Hwee Tou Ng (NUS, Singapore)
- Stefan Riezler (Univ. Heidelberg, Germany)
- Kay Rottmann (Facebook, USA)
- Avneesh Saluja (CMU, USA)
- Wade Shen (MIT-LL, USA)
- Xiaodong Shi (Xiamen Univ., China)
- Stephan Vogel (QCI, USA)
- Taro Watanabe (NICT, Japan)
- Dekai Wu (HKUST, Asia)
- Hao Zhang (Google, USA)
- Jiajun Zhang (CAS, Asia)
- Bing Zhao (SRI, USA)

Acknowledgement

Institute for Multilingual and Multimedia Information

Program of the 10th International Workshop on Spoken Language Translation (IWSLT 2013)

December 05, 2013

08:30 - 09:15	*Registration and welcome coffee*
09.15 - 09:30	Welcome remarks
09:30 - 10:20	Overview of IWSLT 2013 Evaluation
10:20 - 10:40	Coffee Break
10:40 - 11:00	Semantic MT Evaluation with HMEANT for IWSLT 2013 (Dekai Wu)
11:00 - 11:40	ASR System paper: *The 2013 KIT IWSLT Speech-to-Text Systems for German and English* (Kevin Kilgour, Christian Mohr, Michael Heck, Quoc Bao Nguyen, Van Huy Nguyen, Evgeniy Shin, Igor Tseyzer, Jonas Gehring, Markus Müller, Matthias Sperber, Sebastian Stüker and Alex Waibel.)
11:40 - 14:00	*Lunch Break*
14:00 - 14:40	MT System paper: *Edinburgh SLT and MT System Description for the IWSLT 2013 Evaluation* (Alexandra Birch, Nadir Durrani and Philipp Koehn.)
14:40 - 15:00	*Coffee Break*
15:00 - 15:40	SLT System paper: *The RWTH Aachen Machine Translation Systems for IWSLT 2013* (Joern Wuebker, Stephan Peitz, Tamer Alkhouli, Jan-Thorsten Peter, Minwei Feng, Markus Freitag and Hermann Ney.)
15:40 - 16:00	*Coffee Break*
16:00 - 17:30	*Posters (system papers + scientific posters)*
17:30 - 20:00	*Guided Tour Castle of Heidelberg* *Reception: "Glühwein" (Hot Spiced Wine) in the inner courtyard of the castle*
20:00 - 23:30	*Dinner in the Castle*

December 06, 2013

09:30 Beginning

09:30 -10:15 Keynote speech: "*The Human Interpreter in Action – Multilingualism at the European Parliament*" by
Susanne Altenberg, Head of Unit, Directorate for Organisation and Planning, Support to Multilingualism Unit, European Parliament.

10:15 – 10:30 *Coffee Break*

10:30 - 12:00 *Oral Session – Technical Papers: ASR I*

- Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda and Satoshi Nakamura. *Constructing a Speech Translation System using Simultaneous Interpretation Data.*
- Joshua Winebarger, Bao Nguyen, Jonas Gehring, Sebastian Stueker and Alexander Waibel. *The 2013 KIT Quaero Speech-to-Text System for French.*
- Michael Heck, Sebastian Stüker, Sakriani Sakti, Alex Waibel and Satoshi Nakamura. *Incremental Unsupervised Training for University Lecture Recognition.*

12:00 - 13:30 *Lunch Break*

13:30 - 15:30 Oral Session – Technical Papers: MT

- Seppo Enarvi and Mikko Kurimo, *Studies on Training Text Selection for Conversational Finnish Language Modeling.*
- Shachar Mirkin and Nicola Cancedda. *Assessing Quick Update Methods of Statistical Translation Models*
- Teresa Herrmann, Jochen Weiner, Jan Niehues and Alex Waibel. *Analyzing the Potential of Source Sentence Reordering in Statistical Machine Translation*
- Jesús González-Rubio and Francisco Casacuberta. *Improving the Minimum Bayes' Risk Combination of Machine Translation Systems.*

15:30 - 16:00 *Coffee Break*

16:00 - 18:00 Oral Session – Technical papers: MT II

- Christoph Schmidt, Oscar Koller, Hermann Ney, Thomas Hoyoux and Justus Piater. *Using Viseme Recognition to Improve a Signlanguage Translation System*
- Francisco Guzman, Hassan Sajjad, Stephan Vogel and Ahmed Abdelali. *The AMARA Corpus: Building Resources for Translating the Web's Educational Content*.
- Jesús González-Rubio, J.Ramon Navarro-Cerdan and Francisco Casacuberta. *Empirical Study of a Two-Step Approach to Estimate Translation Quality*.
- Li Gong, Aurélien Max and François Yvon. *Improving Bilingual Sub-sentential Alignment by Sampling-based Transpotting*

18:00 Farewell

Keynote

The Human Interpreter in Action – Multilingualism at the European Parliament

Multilingualism is at the heart of the European Parliament. 24 official EU languages make a daunting 552 possible language combinations not to mention many other non-official languages used almost daily (such as the languages of candidates countries, Russian, Arabic, Chinese, Farsi, etc). DG Interpretation and Conferences offers a high quality service for all meetings of the European Parliament and several other EU-institutions and bodies.

In my presentation I will explain how we provide high quality interpretation in a truly multilingual environment and I will try to answer the following questions: What are the main challenges for the interpreters? How to achieve full language coverage? How can new technologies support the work of the interpreters? Man versus Machine or Man and Machine?

Susanne Altenberg is Head of Unit for Multilingualism Support and acting Head of Unit for E-Learning at DG Interpretation and Conferences of the European Parliament. The responsibilities of these units include testing of interpreters and Competitions, grants and support for universities training interpreters, new technologies and the future of interpreting, inter-institutional relations as well as e-learning via Virtual Master Classes and development of speech banks and tools for testing.

Index of Papers

Overview of Evaluation

System Papers

- Krzysztof Wołk and Krzysztof Marasek, **Polish - English Speech Statistical Machine Translation Systems for the IWSLT 2013.**
- M Ali Basha Shaik, Zoltan Tueske, Simon Wiesler, Nussbaum-Thom Markus, Stephan Peitz, Ralf Schlueter and Hermann Ney, **The RWTH Aachen German and English LVCSR systems for IWSLT-2013**.
- Markus Freitag, Stephan Peitz, Joern Wuebker, Hermann Ney, Nadir Durrani, Matthias Huck, Philipp Koehn, Thanh-Le Ha, Jan Niehues, Mohammed Mediani, Teresa Herrmann, Alex Waibel, Nicola Bertoldi, Mauro Cettolo and Marcello Federico, **EU-BRIDGE MT: Text Translation of Talks in the EU-BRIDGE Project**.
- Michaeel Kazi, Tim Anderson, Grant Erdmann, Elizabeth Salesky, Michael Coury, Jessica Ray, Brian Ore, Lane Schwartz, Wade Shen, Jeremy Gwinnup, Michael Hutt, Raymond Slyh, Katherine Young and Terry Gleason, **The MIT-LL/AFRL IWSLT2013 MT system**.
- Ngoc-Quan Pham, Hai-Son Le, Tat-Thang Vu and Chi-Mai Luong, **The Speech Recognition and Machine Translation System of IOIT for IWSLT 2013**.
- Ertuğrul Yılmaz, İlknur Durgar El-Kahlout, Burak Aydın, Zişan Sıla Özil and Coskun Mermer, **TÜBİTAK TURKISH-ENGLISH SUBMISSIONS for IWSLT 2013**.
- Nicola Bertoldi, M. Amin Farajian, Prashant Mathur, Nicholas Ruiz and Marcello Federico, **FBK's Machine Translation Systems for the IWSLT 2013 Evaluation Campaign**.
- Patrick Simianer, Laura Jehl and Stefan Riezler, **The Heidelberg University Machine Translation Systems for IWSLT2013.**
- Peter Bell, Fergus McInnes, Siva Reddy Gangireddy, Mark Sinclair, Alexandra Birch and Steve Renals, **The UEDIN English ASR System for the IWSLT 2013 Evaluation**.
- Sakriani Sakti, Keigo Kubo, Graham Neubig, Tomoki Toda and Satoshi Nakamura, **The NAIST English Speech Recognition System for IWSLT 2013**.
- Thanh-Le Ha, Teresa Herrmann, Jan Niehues, Mohammed Mediani, Eunah Cho, Yuqi Zhang, Isabel Slawik and Alex Waibel, **The KIT Translation Systems for IWSLT 2013**.
- Xingyuan Peng, Xiaoyin Fu, Wei Wei, Zhenbiao Chen and Bo Xu, The CASIA Machine Translation System for IWSLT 2013.

Scientific Papers

- Christoph Schmidt, Oscar Koller, Hermann Ney, Thomas Hoyoux and Justus Piater, **Using Viseme Recognition to Improve a Signlanguage Translation System**.
- Francisco Guzman, Hassan Sajjad, Stephan Vogel and Ahmed Abdelali, **The AMARA Corpus: Building Resources for Translating the Web's Educational Content**.

- Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda and Satoshi Nakamura, **Constructing a Speech Translation System using Simultaneous Interpretation Data**.
- Jesús González-Rubio and Francisco Casacuberta, **Improving the Minimum Bayes' Risk Combination of Machine Translation Systems**.
- Jesús González-Rubio, J.Ramon Navarro-Cerdan and Francisco Casacuberta, **Empirical Study of a Two-Step Approach to Estimate Translation Quality**.
- Joshua Winebarger, Bao Nguyen, Jonas Gehring, Sebastian Stüker and Alexander Waibel, **The 2013 KIT Quaero Speech-to-Text System for French**.
- Li Gong, Aurélien Max and François Yvon, **Improving Bilingual Sub-sentential Alignment by Sampling-based Transpotting**.
- Michael Heck, Sebastian Stüker, Sakriani Sakti, Alex Waibel and Satoshi Nakamura, **Incremental Unsupervised Training for University Lecture Recognition**.
- Seppo Enarvi and Mikko Kurimo, **Studies on Training Text Selection for Conversational Finnish Language Modeling**.
- Shachar Mirkin and Nicola Cancedda, **Assessing Quick Update Methods of Statistical Translation Models**.
- Teresa Herrmann, Jochen Weiner, Jan Niehues and Alex Waibel, **Analyzing the Potential of Source Sentence Reordering in Statistical Machine Translation**.
- Eunah Cho, Thanh-Le Ha and Alex Waibel, **CRF-based Disfluency Detection using Semantic Features for German to English Spoken Language Translation**.
- Evgeniy Shin, Sebastian Stüker, Kevin Kilgour, Christian Fügen and Alex Waibel, **Maximum Entropy Language Modeling for Russian ASR**.
- Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch and Sanjeev Khudanpur, **Improved Speech-to-Text Translation with the Fisher and Callhome Spanish–English Speech Translation Corpus**.
- Markus Saers and Dekai Wu, **Unsupervised Learning of Bilingual Categories in Inversion Transduction Grammar Induction**.
- Benjamin Marie and Aurélien Max, **A Study in Greedy Oracle Improvement of Translation Hypotheses**.
- Sankaranarayanan Ananthakrishnan, Wei Chen, Rohit Kumar and Dennis Mehay, **Source-Error Aware Phrase-Based Decoding for Robust Conversational Spoken Language Translation**.

- 1Akiko Sakamoto, Kazuhiko Abe, Kazuo Sumita and Satoshi Kamatani, **Evaluation of a Simultaneous Interpretation System and Analysis of Speech Log for User Experience Assessment**.
- Shahab Jalalvand and Daniele Falavigna, **Parameter Optimization for Iterative Confusion Network Decoding in Weather-Domain Speech Recognition**.

Report on the 10th IWSLT Evaluation Campaign

Mauro Cettolo[(1)] Jan Niehues[(2)] Sebastian Stüker[(2)] Luisa Bentivogli[(1)] Marcello Federico[(1)]

[(1)] FBK - Via Sommarive 18, 38123 Trento, Italy
[(2)] KIT - Adenauerring 2, 76131 Karlsruhe, Germany

Abstract

The paper overviews the tenth evaluation campaign organized by the IWSLT workshop. The 2013 evaluation offered multiple tracks on lecture transcription and translation based on the TED Talks corpus. In particular, this year IWSLT included two automatic speech recognition tracks, on English and German, three speech translation tracks, from English to French, English to German, and German to English, and three text translation track, also from English to French, English to German, and German to English. In addition to the official tracks, speech and text translation optional tracks were offered involving 12 other languages: Arabic, Spanish, Portuguese (B), Italian, Chinese, Polish, Persian, Slovenian, Turkish, Dutch, Romanian, Russian. Overall, 18 teams participated in the evaluation for a total of 217 primary runs submitted. All runs were evaluated with objective metrics on a current test set and two progress test sets, in order to compare the progresses against systems of the previous years. In addition, submissions of one of the official machine translation tracks were also evaluated with human post-editing.

1. Introduction

This paper overviews the results of the evaluation campaign organized by the International Workshop of Spoken Language Translation. The IWSLT evaluation has been now running for a decade and has offered along these years a variety of speech translation tasks [1, 2, 3, 4, 5, 6, 7, 8, 9]. The 2013 IWSLT evaluation continued along the line set in 2010, by focusing on the translation of TED Talks, a collection of public speeches covering many different topics. As in the previous two years, the evaluation included tracks for all the core technologies involved in the spoken language translation task, namely:

- Automatic speech recognition (ASR), i.e. the conversion of a speech signal into a transcript,
- Machine translation (MT), i.e. the translation of a polished transcript into another language,
- Spoken language translation (SLT), that addressed the conversion and translation of a speech signal into a transcript in another language.

However, with respect to previous rounds, new languages have been added to each track. The ASR track included besides English also German, and the SLT and MT track offered English-French, English-German, and German-English translation directions. Besides the official evaluation tracks, many other optional translation directions were also offered. Optional SLT directions were from English to Spanish, Portuguese (B), Italian, Chinese, Polish, Slovenian, Arabic, and Persian. Optional MT translation directions were: English from/to Arabic, Spanish, Portuguese (B), Italian, Chinese, Polish, Persian, Slovenian, Turkish, Dutch, Romanian, and Russian. For each official and optional translation direction, training and development data were supplied by the organizers through the workshop's website. Major parallel collections made available to the participants were the WIT^3 [10] corpus of TED talks, all data from the WMT 2013 workshop (CITE), the MULTIUN corpus (CITE), and the SETimes parallel corpus (CITE). A list of monolingual resources was provided too, that includes both freely available corpora and corpora available from the LDC. Test data were released at the begin of each test period, requiring participants to return one primary run and optional contrastive runs within one week. The schedule of the evaluation was organized as follows: June 8, release of training data; Sept 2-8, ASR test of period; Sept 9-15, SLT test period; Oct 7-13, MT test period; Oct 7-20, test period of all optional directions.

All runs submitted by participants were evaluated with automatic metrics. In addition, MT runs of the English-French direction were evaluated manually. While in the past years SLT and MT outputs were evaluated through subjective rankings, this year another method was investigated. In particular, we tried to address the utility of MT output by measuring the post-editing effort needed by a professional translator to fix it.

This year, 18 participant sites registered (see Table 1) submitting a total of 217 primary runs: 28 to the ASR track, 10 to the SLT track, and 179 to the MT track (see Sections 3.3, 4.3, 5.3 for details).

In the rest of the paper we first outline the main goals of the IWSLT evaluation and then each single track in detail, in particular: its specifications, supplied language resources, evaluation methods, and results. The paper ends with some concluding remarks about the experience made in this evaluation exercise, followed by appendixes that complement the information given in the specific sections.

2. TED Talks

2.1. TED events

The translation of TED talks was introduced for the first time at IWSLT 2010. TED is a nonprofit organization that "invites the world's most fascinating thinkers and doers [...] to give the talk of their lives". Its website[1] makes the video recordings of the best TED talks available under the Creative Commons license. All talks have English captions, which have also been translated into many languages by volunteers worldwide. In addition to the official TED events held in North America, a series of independent TEDx events are regularly held around the world, which share the same format of the original TED talks but are hold in the language of the hosting country. Recently, an effort was made to set up a web repository [10] that distributes dumps of the available TED talks transcripts and translations under form of parallel texts, ready to use for training and evaluating MT systems. At this time, parallel data between English and 15 foreign languages are available in addition to evaluation sets results achieved by baseline MT systems trained for each translation direction.

Besides representing a popular benchmark for spoken language technology, the TED Talks task embeds interesting research challenges which are unique among the available speech recognition and machine translation benchmarks. TED Talks is a collection of rather short speeches (max 18 minutes each, roughly equivalent to 2,500 words) which cover a wide variety of topics. Each talk is delivered in a brilliant and original style by a very skilled speaker and, while addressing a wide audience, it pursues the goal of both entertaining and persuading the listeners on a specific idea. From the point of view of ASR, TED talks require copying with background noise – e.g. applauses and laughs by the public –, different accents including non native speakers, varying speaking rates, prosodic aspects, and, finally, narrow topics and personal language styles. From an application perspective, TED Talks transcription is the typical life captioning scenario, which requires producing polished subtitles in real-time.

From the point of view of machine translation, translating TED Talks implies dealing with spoken rather than written language, which is hence expected to be structurally less complex, formal and fluent. Moreover, as human translations of the talks are required to follow the structure and rythm of the English captions [2], a lower amount of rephrasing and reordering is expected than in ordinary translation of written documents.

From an application perspective, TED Talks suggest translation tasks ranging from off-line translation of written captions, up to on-line speech translation, requiring a tight integration of MT with ASR possibly handling stream-based processing.

[1] http://www.ted.com

[2] See recommendations to translators in http://translations.ted.org/wiki.

3. ASR Track

3.1. Definition

The goal of the *Automatic Speech Recognition* (ASR) track for IWSLT 2013 was to transcribe English TED talks and German TEDx talks. The speech in TED lectures is in general planned, well articulated, and recorded in high quality. The main challenges for ASR in these talks are to cope with a large variability of topics, the presence of non-native speakers, and the rather informal speaking style. For the German TEDx talks the recording conditions are a little bit more difficult than for the English TED talks. While the TEDx talks aim to mimic the TED talks, they are not as well prepared and well rehearsed as the TED lectures, and recording is often done by amateurs resulting in often worse recording quality than the TED lectures.

The result of the recognition of the talks is used for two purposes. It is used to measure the performance of ASR systems on the talks and it is used as input for the spoken language translation evaluation (SLT), see Section 4.

3.2. Evaluation

Participants had to submit the results of the recognition of the tst2013 set in CTM format. The word error rate was measured case-insensitive. After the end of the evaluation a first scoring was performed with the first set of references. This was followed by an adjudication phase in which participants could point out errors in the reference transcripts. The adjudication results were collected and combined into the final set of references with which the official score were calculated.

In order to measure the progress of the systems over the years on English, participants also had to provide results on the test sets from 2011 and 2012, i.e. tst2011 and tst2012.

3.3. Submissions

For this year's evaluation we received primary submissions from eight sites: all of which participated in the English ASR task and four also in the German ASR task. For English we further received a total of nine contrastive submissions from six sites. For German we received eight contrastive submissions from three sites.

3.4. Results

The detailed results of the primary submissions of the evaluation in terms of word error rate (WER) can be found in Appendix A.1. The word error rate of the submitted systems in in the range of 13.5%-27.2% for English and 25.2%-37.8% for German.

In German, the fact that TEDx have sometimes worse recording conditions than TED talks was reflected by the fact that one talk in the German tst2013 had WERs above 80%, due to a bad recording set-up with high noise. All other WERs were mostly below 30% and 20%, for two talks even below 10%.

Table 1: List of Participants

NTT-NAIST	NTT Communication Science Labs, Japan & NAIST[11]
KIT	Karlsruhe Institute of Technology, Germany [12, 13]
RWTH	Rheinisch-Westfälische Technische Hochschule Aachen, Germany [14, 15]
EU-BRIDGE	RWTH& UEDIN& KIT& FBK[16]
HDU	Dept. of Computational Linguistics, Heidelberg University, Germany [17]
UEDIN	University of Edinburgh, UK [18, 19, 20]
FBK	Fondazione Bruno Kessler, Italy [21, 22]
PRKE-IOIT	Inst. of Inform. and Techn., Vietnamese Academy of Science and Technology [23]
POSTECH	Pohang University of Science and Technology, Korea [24]
MITLL-AFRL	Mass. Institute of Technology/Air Force Research Lab., USA [25]
QCRI	Qatar Computing Research Institute, Qatar Foundation, Qatar [26]
MSR-FBK	Microsoft Corporation, USA, and FBK[27]
HKUST	Hong Kong University of Science and Technology, Hong Kong [28]
NICT	National Institute of Communications Technology, Japan [29, 30]
NAIST	Nara Institute of Science and Technology, Japan [31]
PJIIT	Polish-Japanese Institute of Information Technology, Poland [32]
CASIA	Institute of Automation, Chinese Academy of Sciences, China [33]
TUBITAK	TUBITAK - Center of Research for Advanced Technologies, Turkey

For English, it can be seen that all participants from IWSLT2011 and IWSLT2012 made significant progresses over the years, e.g., bringing down the WER from 13.5% to 7.9% on tst2011, a relative reduction by 41% over the course of three years.

4. SLT Track

4.1. Definition

The SLT track required participants to translate the English and German talks of tst2013 from the audio signal (see Section 3). The challenge of this translation task over the MT track is the necessity to deal with automatic, and in general error prone, transcriptions of the audio signal, instead of correct human transcriptions.

For German, participants had to translate into English. For English as source language, participants had to translate into French and German. In addition, participants could also optionally translate from English into one of the following languages: Arabic, Spanish, Farsi, Italian, Polish, Brazilian Portuguese, Slovenian, and Mandarin Chinese.

4.2. Evaluation

For the evaluation, participants could choose to either use their own ASR technology, or to use ASR output provided by the confer

ence organizers. In order to facilitate scoring, participants had to to segment the audio according to the manual reference segmentation provided by the organizers of the evaluation.

For English, the ASR output provided by the organizers was a ROVER combination of the output from five submissions to the ASR track. The result of the ROVER had a WER of 12.4%. For German we used the output from KIT, as ROVER combination with other systems did not give any performance gains, and the German KIT ASR system scored best before the end of the adjudication.

The results of the translation had to be submitt ed in the same format as for the machine translation track (see Section 5).

4.3. Submissions

We received ten primary and nine contrastive submissions from five participants, English to French receiving the most submissions. In English to Arabic and English to Chinese only one participant each submitted results.

4.4. Results

The detailed results of the automatic evaluation in terms of BLEU and TER can be found in Appendix A.1. Appendix A.2 contains the results of the progress test set for English to French.

5. MT Track

5.1. Definition

The MT TED track basically corresponds to a subtitling translation task. The natural translation unit considered by the human translators volunteering for TED is indeed the single caption — as defined by the original transcript — which in general does not correspond to a sentence, but to fragments of it that fit the caption space. While translators can look at the context of the single captions, arranging the MT task in this way would make it particularly difficult, especially when word re-ordering across consecutive captions occurs. For this

Table 2: Monolingual resources for official language pairs

data set	lang	sent	token	voc
train	De	146k	2.66M	107.4k
	En	159k	3.20M	58.3k
	Fr	158k	3.36M	70.7k

reason, we preprocessed all the parallel texts to re-build the original sentences, thus simplifying the MT task.

As already stated in the Introduction, for each official and optional translation direction, in-domain training and development data were supplied through the website of the WIT[3] [10], while out-of-domain training data through the workshop's website. With respect to edition 2012 of the evaluation campaign, some of the talks added to the TED repository during the last year have been used to define the new evaluation sets (tst2013), while the remaining talks have been included in the training sets. For reliably assessing progress of MT systems over the years, the evaluation sets of editions 2011 and 2012 were distributed together with tst2013 as progressive test sets, when available. Development sets (dev2010 and tst2010) are either the same of past editions or have been built upon the same talks.

With respect to all the other directions, the $DeEn$ MT task is an exception; in fact, its dev2012 and tst2013 - development and evaluation sets, respectively - derives from those prepared for the ASR/SLT tracks, which consist of TEDX talks delivered in German language; therefore, no overlap exists with any other TED talk involved in other tasks. Anyway, the standard dev2010 and tst2010 development sets have been released as well.

Tables 2 and 3 provides statistics on in-domain texts supplied for training, development and evaluation purposes for the official directions.

Reference results from baseline MT systems on the development set tst2010 are provided via the WIT3 repository. This helps participants and MT scientists to assess their experimental outcomes.

MT baselines were trained from TED data only, i.e. no additional out-of-domain resources were used. The standard tokenization via the tokenizer script released with the Europarl corpus [34] was applied to all languages, with the exception of Chinese and Arabic languages, which were preprocessed by, respectively: the Stanford Chinese Segmenter [35]; either AMIRA [36], in the Arabic-to-English direction, or the QCRI-normalizer,[3] in the English-to-Arabic direction.

The baselines were developed with the Moses toolkit. Translation and lexicalized reordering models were trained on the parallel training data; 5-gram LMs with improved Kneser-Ney smoothing were estimated on the target side of the training parallel data with the IRSTLM toolkit. The weights of the log-linear interpolation model were optimized on dev2010 with the MERT procedure provided with Moses.

Table 3: Bilingual resources for official language pairs

task	data set	sent	tokens source	tokens target	talks
MT$_{EnFr}$	train	154k	3.06M	3.27M	1169
	dev2010	887	20,1k	20,2k	8
	tst2010	1,664	32,0k	33,9k	11
	tst2011	818	14,5k	15,6k	8
	tst2012	1,124	21,5k	23,5k	11
	tst2013	1,026	21,7k	23,3k	16
MT$_{DeEn}$	train	139k	2.59M	2.75M	1064
	dev2010	887	19,1k	20,1k	8
	tst2010	1,565	30,3k	32,0k	11
	dev2012	1,165	20,8k	21,6k	7
	tst2013	1,369	22,4k	22,8k	9
MT$_{EnDe}$	train	139k	2.75M	2.59M	1064
	dev2010	887	20,1k	19,1k	8
	tst2010	1,565	32,0k	30,3k	11
	tst2011	1,436	27,1k	26,4k	16
	tst2012	1,704	30,8k	29,3k	15
	tst2013	993	20,9k	19.7k	16

5.2. Evaluation

The participants to the MT track had to provide the results of the translation of the test sets in NIST XML format. The output had to be true-cased and had to contain punctuation.

The quality of the translations was measured automatically against the human translations created by the TED open translation project, and by human subjective evaluation (Section 5.5).

The evaluation specifications for the MT track were defined as case-sensitive with punctuation marks (case+punc). Tokenization scripts were applied automatically to all run submissions prior to evaluation.

Evaluation scores were calculated for the two automatic standard metrics BLEU and TER, as implemented in mteval-v13a.pl[4] and tercom-0.7.25[5], respectively.

5.3. Submissions

We received 68 submissions from 15 different sites, distributed as follows: 20 for the three official language pairs, 48 on optional directions.

The pairs that attracted the most interest are the official pairs – seven each for EnFr and DeEn, six for EnDe – and those involving Chinese (a total of nine in the two directions), Arabic (seven), Farsi (five) and Russian (five). Each pair received at least one submission.

The total number of primary runs, on evaluation set tst2013 and on progressive test sets tst2011 and tst2012, is

[3] Specifically developed for IWSLT 2013 by P. Nakov and F. Al-Obaidli at Qatar Computing Research Institute.

[4] http://www.itl.nist.gov/iad/mig/tests/mt/2009/

[5] http://www.cs.umd.edu/ snover/tercom/

179; in addition, we were asked to evaluate also 156 contrastive runs.

5.4. Results

Table 4: BLEU and TER scores of baseline SMT systems on tst2013 for all language pairs. (⋆) Char-level scores.

pair		direction → BLEU	direction → TER	direction ← BLEU	direction ← TER
En	Fr	31.94	48.59	–	–
	De	19.58	59.81	19.07	65.94
	Ar	12.12	68.73	22.71	59.02
	Es	29.01	50.99	33.18	45.58
	Fa	8.94	72.74	12.17	88.88
	It	26.59	52.75	30.82	50.35
	Nl	22.82	57.66	28.00	54.49
	Pl	10.31	76.16	16.31	67.33
	Pt	29.65	46.85	35.80	42.93
	Ro	16.18	68.29	24.85	54.21
	Ru	13.69	71.30	18.57	64.99
	Sl	9.49	72.16	14.62	69.70
	Tr	6.62	79.96	12.24	75.90
	Zh	⋆18.15	⋆72.34	12.29	70.60

First of all, for reference purposes Table 4 shows BLEU and TER scores on the tst2013 evaluation sets of the baseline systems we developed as described in Section 5.1.

The results on the official test set for each participant are shown in Appendix A.1. For most languages, we show the case-sensitive and case-insensitive BLEU and TER scores. In contrast to the other language pairs, in the German to English translation task the source contained disfluencies. Therefore, the translation are evaluated once against translation containing disfluencies and once against reference containing no disfluencies. Furthermore, for English to Chinese we report character-level and word-level scores.

These results also show again the scores of the baseline system. Thereby, it is possible to see the improvements of the submitted systems on the different languages over the baseline system. The largest improvements could be gained on Slovenian-English by 9.44 BLEU points.

In Appendix A.2 the results on the progress test sets test2011 and test2012 are shown. When comparing the results to the submissions from last year, the performance could be improved in nearly all tasks.

5.5. Human Evaluation

Human evaluation was carried out on all primary runs submitted by participants to one of the official tracks of the TED task, namely the *official* MT English-French track.

This year's human evaluation saw the introduction of a major novelty. In fact, the traditional *Relative Ranking* task was substituted by a *Post-Editing* task and, accordingly, HTER (Human-mediated Translation Edit Rate) was adopted as the official evaluation metrics to rank the systems.

Post-Editing, i.e. the manual correction of machine translation output, has long been investigated by the translation industry as a form of machine assistance to reduce the costs of human translation. Nowadays, Computer-aided translation (CAT) tools incorporate post-editing functionalities, and a number of studies [37, 38] demonstrate the usefulness of MT to increase professional translators' productivity. The MT TED task offered in IWSLT can be seen as an interesting application scenario to test the utility of MT systems in a real subtitling task.

From the point of view of the evaluation campaign, our goal was to adopt a human evaluation framework able to maximize the benefit to the research community, both in terms of information about MT systems and data and resources to be reused. With respect to traditional judgments of translation quality (i.e. adequacy/fluency and ranking tasks), the post-editing task has the double advantage of producing *(i)* a set of edits pointing to specific translation errors, and *(ii)* a set of additional reference translations. Both these byproducts are very useful for MT system development and evaluation. Furthermore, HTER[39] - which consists of measuring the minimum edit distance between the machine translation and its manually post-edited version - has been shown to correlate quite well with human judgments of MT quality.

The human evaluation setup and the collection of post-editing data are presented in Section 5.5.1, whereas the results of the evaluation are presented in Section 5.5.2.

5.5.1. Evaluation Setup and Data Collection

All 2013 systems participating in the English-French MT track were manually evaluated on a subset of the 2012 progress test set *(tst2012)*[6]. The Human Evaluation (HE) set represents around the initial 50% of each of the 11 *tst2012* talks, for a total of 580 segments and around 10,000 words. This choice of selecting a consecutive block of sentences for each talk was determined by the need of realistically simulating a caption post-editing task on several TED talks.

In order to evaluate the MT systems, the *bilingual* post-editing task was chosen, where professional translators are required to post-edit the MT output directly according to the source sentence. Bilingual post-editing is expected to give more accurate results than monolingual post-editing as post-editors do not depend on an given - and possibly imprecise - translation.

As far as evaluation metrics are concerned, HTER [39] is a semi-automatic metric derived from TER (Translation Edit Rate). TER measures the amount of editing that a human would have to perform to change a machine translation so that it exactly matches a given reference translation. HTER

[6]Since all the data produced for human evaluation will be made publicly available thorough the WIT^3 repository, we used the 2012 test set in order to keep the 2013 test set blind to be used as a progress test for next year's evaluation.

is a variant of TER where a new reference translation is generated by applying the minimum number of post-edits to the given MT output. This new *targeted* reference is then used as the only reference translation to calculate the MT output TER.

In the preparation of the data to be collected, some constraints were identified to ensure the soundness of the evaluation of the seven systems participating in the task: *(i)* each translator must post-edit all segments of the HE set, *(ii)* each translator must post-edit the segments of the HE set only once, and *(iii)* each MT system must be equally post-edited by all translators.

Given that we had seven systems to evaluate, in order to satisfy the above constraints we resorted to seven professional translators. Moreover, in order to cope with variability of post-editors (i.e. some translators could systematically post-edit more than others) we devised a scheme that dispatches MT outputs to translators both randomly and satisfying the uniform assignment constraints. Seven documents were hence prepared including all source segments of the HE set and, for each source segment, one MT output selected from one of the seven systems.

Documents were delivered to a language service provider together with instructions to be passed on to the translators, and the post-editing tasks were run using the tool developed under the MateCat project[7], an enterprise-level CAT tool. Both the post-editing interface and the guidelines given to translators are presented in Appendix B.

The resulting collected data consist of seven new reference translations for each of the 580 sentences of the HE set. Each one of these seven references represents the targeted translation of the system output from which it was derived. From the point of view of the system output, one targeted translation and other six untargeted translations are available.

Table 5 shows information about the characteristics of the work carried out by post-editors. First, the post-editing effort for each translator is given. Post-editing effort is to be interpreted as the number of actual edit operations performed to produce the post-edited version and - consequently - it is calculated as the HTER of all the system sentences post-edited by each single translator. As we can see from the table, PE effort is highly variable among post-editors, ranging from 19.51% to 42.60%. Data about standard deviation confirm post-editor variability, showing that the seven translators produced quite different post-editing effort distributions.

To further study post-editor variability, we exploited the official reference translations available for this TED track and we calculated the TER of the outputs assigned to each translator for post-editing (*Sys TER* Column in Table 5), as well as the related standard deviation.

As we can see from the table, the documents presented to translators (composed of segments produced by different systems) are very homogeneous, as they show very similar TER scores and standard deviation figures. This also confirms that the procedure followed in data preparation was effective.

[7] www.matecat.com

Table 5: Post-editing information for each Post-editor

PEditor	PE Effort	*std-dev*	Sys TER	*std-dev*
PE 1	24.93	17.74	40.27	20.32
PE 2	34.03	19.86	39.48	19.89
PE 3	42.60	22.47	40.61	20.19
PE 4	32.78	21.07	39.98	20.97
PE 5	19.51	15.55	40.82	20.95
PE 6	30.64	19.48	40.42	20.70
PE 7	34.60	23.92	39.39	20.62

The variability observed in post-editing effort - despite the similarity of the input documents - is most probably due to translators' subjectivity in carrying out the post-editing task. Thus, post-editor variability ai an issue to be addressed to ensure a sound evaluation of the systems.

5.5.2. Evaluation Results

As seen in the previous section, being able to reduce post-editors' variability would allow a more reliable and consistent evaluation of MT systems. To this purpose, the HTER for each system submission was calculated under two different settings, namely *(i)* using the targeted reference only (*Tgt PEref* setting), and *(ii)* using all the seven references produced by all the post-editors for each sentence (*All PErefs* setting).

The scores resulting from the application of the two HTER settings are shown in Table 6, which also presents a comparison of HTER scores and rankings with those obtained using the related automatic metrics TER[8].

Table 6: Official human evaluation results and comparisons with other metrics

System Ranking	**HTER** ***HE Set all PErefs***	**HTER** HE Set Tgt PEref	**TER** HE Set ref	**TER** Test Set ref
EU-BRIDGE	**18.67**	29.83	38.71	38.72
KIT	**20.01**	*29.64*	39.20	39.22
UEDIN	**20.69**	31.61	39.81	39.83
RWTH	**21.06**	31.64	*39.70*	39.95
FBK	**21.41**	32.29	40.38	40.56
MITLL-AFRL	**22.24**	32.31	41.37	41.47
PRKE-IOIT	**22.26**	*32.01*	41.81	41.52
Rank Corr.		.857	.964	1.00

As shown in the table, the HTER reduction obtained in the *All PErefs* setting (Column 2) with respect to the *Tgt PEref* setting (Column 3) clearly shows that exploiting all the available reference translations is a viable way to control and overcome post-editors' variability, obtaining an HTER

[8] Note that since HTER and TER are edit-distance measures, lower numbers indicate better performances

which is more informative about the real performances of the systems. This is also confirmed by the range of standard deviations observed for the scores of the systems, which for *Tgt PEref* ranges from 20.57 to 23.18, while for *All PEref* ranges from 12.84 to 14.31.

For this reason, the scores and overall ranking of the systems as resulting in the *All PErefs* setting have been chosen as the official results of human evaluation.

In general, the very low HTER results obtained demonstrate that the overall quality of the systems is very high. Moreover, all systems are very close to each other. To establish the reliability of system ranking, for all pairs of systems we calculated the statistical significance of the observed differences in performance. Statistical significance was assessed with the *approximate randomization* method [40], a statistical test well-established in the NLP community [41] and that, especially for the purpose of MT evaluation, has been shown [42] to be less prone to type-I errors than the bootstrap method [43]. According to the approximate randomization test based on 10,000 iterations, a winning system cannot be indicated, as there is no system that is significantly better than all other systems. Significant differences can be found only between the top-scoring system (EU-BRIDGE) and the three bottom-scoring ones. In particular, significance with respect to FBK is at $p \leq 0.1$, while significance with respect to MITLL-AFRL and PRKE-IOIT is at $p \leq 0.05$.

A number of additional observations can be drawn by comparing the official results with results obtained with other metrics (Columns 3,4,5 in Table 6).

In general, HTER reduces the edit rate with respect to TER. More specifically, we can see a reduction of around 25% for HTER calculated with only one targeted reference(*Tgt PEref* setting), and of around 50% for HTER calculated with all post-edited references (*All PErefs* setting).

Moreover, the correlation between evaluation metrics is measured using *Spearman's rank correlation coefficient* $\rho \in$ [-1.0, 1.0], with ρ = 1.0 if all systems are ranked in same order, ρ = -1.0 if all systems ranked in reverse order and ρ = 0.0 if no correlation exists. We can see from Table 6 that completely automatic metrics (TER) correlate well with the official HTER. In particular, TER calculated on the whole 2012 test set correlates perfectly, confirming that automatic metrics are more reliable when the quantity of evaluation data increases.

To conclude, the post-editing task introduced this year for manual evaluation brought benefit to the IWSLT community, and in general to the MT field. In fact, producing post-edited versions of all the participating systems' outputs allowed us to carry out a quite informative evaluation by minimizing the variability of post-editors, who naturally tend to diverge from the post-editing guidelines and personalize their translations. Moreover, a number of additional reference translations will be available for further development and evaluation of MT systems.

6. Conclusions

We have reported on the evaluation campaign organized for the tenth edition of the IWSLT workshop. The evaluation has addressed three tracks: automatic speech recognition of talks (in English and German), speech-to-text translation, and text-to-text translation, both from German to English, English to German, and English to French. Besides the official translation directions, many optional translation tasks were available, too, including 12 additional languages. For each task, systems had to submit runs on three different test sets: a newly created official test set, and two progress test sets created and used for the 2012 and 2011 evaluations, respectively. This year, 18 participants took part in the evaluation, submitting a total of 217 primary runs, which were all scored with automatic metrics. We also manually evaluated runs of the English-French text translation track. In particular, we asked professional translators to post-edit all system outputs on a subset of the 2012 progress test set, in order to produce *close references* for them. While we have observed a significant variability among translators, in terms of post-edit effort, we could obtain more reliable scores by using all the produced post-edits as reference translations. By using the HTER metric, the post-edit effort of the best performing system results remarkably low, namely less than 19%. Considering that this is still an upper bound of the ideal HTER score, this percentage of post-editing seems to be another strong argument supporting the utility of machine translation for human translators.

7. Acknowledgements

Research Group 3-01' received financial support by the '*Concept for the Future*' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative. The work leading to these results has received funding from the European Union under grant agreement no 287658 — Bridges Across the Language Divide (EU-BRIDGE).

8. References

[1] Y. Akiba, M. Federico, N. Kando, H. Nakaiwa, M. Paul, and J. Tsujii, "Overview of the IWSLT04 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2004, pp. 1–12.

[2] M. Eck and C. Hori, "Overview of the IWSLT 2005 evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Pittsburgh, PA, 2005, pp. 1–22.

[3] P. Michael, "Overview of the IWSLT 2006 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2006, pp. 1–15.

[4] C. S. Fordyce, "Overview of the IWSLT 2007 evalu-

ation campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Trento, Italy, 2007, pp. 1–12.

[5] M. Paul, "Overview of the IWSLT 2008 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Waikiki, Hawaii, 2008, pp. 1–17.

[6] ——, "Overview of the IWSLT 2009 Evaluation Campaign," in *Proceedings of the sixth International Workshop on Spoken Language Translation*, Tokyo, Japan, 2009, pp. 1–18.

[7] M. Paul, M. Federico, and S. Stüker, "Overview of the IWSLT 2010 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Paris, France, 2010, pp. 3–27.

[8] M. Federico, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2011 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, San Francisco, USA, 2011, pp. 11–27.

[9] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Hong Kong, HK, 2012, pp. 11–27.

[10] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks," in *Proceedings of the Annual Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012. [Online]. Available: http://hltshare.fbk.eu/EAMT2012/html/Papers/59.pdf

[11] K. Sudoh, G. Neubig, K. Duh, and H. Tsukada, "NTT-NAIST SMT Systems for IWSLT 2013," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[12] K. Kilgour, C. Mohr, M. Heck, Q. B. Nguyen, V. H. Nguyen, E. Shin, I. Tseyzer, J. Gehring, M. M̈ller, M. Sperber, S. Stüker, and A. Waibel, "The 2013 KIT IWSLT Speech-to-Text Systems for German and English," in *Proceedings of the tenth International Workshop on Spoken Language Trans lation (IWSLT)*, Heidelberg, Germany, 2013.

[13] T.-L. Ha, T. Herrmann, J. Niehues, M. Mediani, E. Cho, Y. Zhang, I. Slawik, and A. Waibel, "The KIT Translation Systems for IWSLT 2013," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[14] J. Wuebker, S. Peitz, T. Alkhouli, J.-T. Peter, M. Feng, M. Freitag, and H. Ney, "The RWTH Aachen Machine Translation Systems for IWSLT 2013," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[15] M. A. B. Shaik1, Z. Tüske, S. Wiesler, M. Nušbaum-Thom, S. Peitz, R. Schl̈er, and H. Ney, "The rwth aachen german and english lvcsr systems for iwslt-2013," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[16] M. Freitag, S. Peitz, J. Wuebker, H. Ney, N. Durrani, M. Huck, P. Koehn, T.-L. Ha, J. Niehues, M. Mediani, T. Herrmann, A. Waibel, N. Bertoldi, M. Cettolo, and M. Federico, "EU-BRIDGE MT: Text Translation of Talks in the EU-BRIDGE Project," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[17] P. Simianer, L. Jehl, and S. Riezler, "The Heidelberg University Machine Translation Systems for IWSLT2013," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[18] P. Bell, F. McInnes, S. R. Gangireddy, M. Sinclair, A. Birch, and S. Renals, "The UEDIN English ASR System for the IWSLT 2013 Evaluation," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[19] J. Driesen, P. Bell, M. Sinclair, and S. Renals, "Description of the uedin system for german asr," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[20] A. Birch, N. Durrani, and P. Koehn, "Edinburgh SLT and MT System Description for the IWSLT 2013 Evaluation," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[21] D. Falavigna, R. Gretter, F. Brugnara, D. Giuliani, and R. H. Serizel, "FBK @ IWSLT 2013 - ASR tracks," in *Proceedings of the tenth International Workshop on Spoken Language Trans lation (IWSLT)*, Heidelberg, Germany, 2013.

[22] N. Bertoldi, M. A. Farajian, P. Mathur, N. Ruiz, and M. Federico, "FBK's Machine Translation Systems for the IWSLT 2013 Evaluation Campaign," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[23] N.-Q. Pham, H.-S. Le, T.-T. Vu, and C.-M. Luong, "The Speech Recognition and Machine Translation System of IOIT for IWSLT 2013," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[24] H. Na and J.-H. Lee, "A Discriminative Reordering Parser for IWSLT 2013," in *Proceedings of the tenth International Workshop on Spoken Language Trans lation (IWSLT)*, Heidelberg, Germany, 2013.

[25] M. Kazi, M. Coury, E. Salesky, J. Ray, W. Shen, T. Gleason, T. Anderson, G. Erdmann, L. Schwartz, B. Ore, R. Slyh, J. Gwinnup, K. Young, and M. Hutt, "The MIT-LL/AFRL IWSLT-2013 MT Systems," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[26] H. Sajjad, F. Guzmán, P. Nakov, A. Abdelali, K. Murray, F. A. Obaidli, and S. Vogel, "QCRI at IWSLT 2013: Experiments in Arabic-English and English-Arabic Spoken Language Translation," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[27] A. Aue, Q. Gao, H. Hassan, X. He, G. Li, N. Ruiz, and F. Seide, "MSR-FBK IWSLT 2013 SLT System Description," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[28] C. kiu Lo, M. Beloucif, and D. Wu, "Improving machine translation into Chinese by tuning against Chinese MEANT," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[29] C.-L. Huang, P. R. Dixon, S. Matsuda, Y. Wu, X. Lu, M. Saiko, and C. Hori, "The nict asr system for iwslt 2013," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[30] A. Finch, O. Htun, and E. Sumita, "The NICT Translation System for IWSLT 2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[31] S. Sakti, K. Kubo, G. Neubig, T. Toda, and S. Nakamura, "The naist english speech recognition system for iwslt 2013," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[32] K. Wolk and K. Marasek, "Polish - englishspeechstatistical machine translationsystems for the iwslt 2013," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[33] X. Peng, X. Fu, W. Wei, Z. Chen, W. Chen, and B. Xu, "The casia machine translation system for iwslt 2013," in *Proceedings of the tenth International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[34] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, Phuket, Thailand, September 2005, pp. 79–86.

[35] H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning, "A conditional random field word segmenter," in *Fourth SIGHAN Workshop on Chinese Language Processing*, 2005.

[36] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks," in *HLT-NAACL 2004: Short Papers*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 149–152.

[37] M. Federico, A. Cattelan, and M. Trombetti, "Measuring user productivity in machine translation enhanced computer assisted translation," in *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*, 2012. [Online]. Available: http://www.mt-archive.info/AMTA-2012-Federico.pdf

[38] S. Green, J. Heer, and C. D. Manning, "The efficacy of human post-editing for language translation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 439–448.

[39] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proceedings of the The Seventh Conference of the Association for Machine Translation in the Americas (AMTA)*, Cambridge, USA, 2006, pp. 223–231.

[40] E. W. Noreen, *Computer Intensive Methods for Testing Hypotheses: An Introduction*. Wiley Interscience, 1989.

[41] N. Chinchor, L. Hirschman, and D. D. Lewis, "Evaluating message understanding systems: An analysis of the third message understanding conference (muc-3)," *Computational Linguistics*, vol. 19, no. 3, pp. 409–449, 1993.

[42] S. Riezler and J. T. Maxwell, "On some pitfalls in automatic evaluation and significance testing for

MT," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 57–64. [Online]. Available: http://www.aclweb.org/anthology/W/W05/W05-0908

[43] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Chapman and Hall, 1993.

Appendix A. Automatic Evaluation

"case+punc" evaluation : case-sensitive, with punctuations tokenized
"no_case+no_punc" evaluation : case-insensitive, with punctuations removed

A.1. Official Testset (*tst2013*)

· All the sentence IDs in the IWSLT 2012 testset were used to calculate the automatic scores for each run submission.
· ASR and MT systems are ordered according to the *WER* and *BLEU* metrics, respectively.
· All automatic evaluation metric scores are given as percent figures (%).

TED : ASR English (ASR$_{EN}$)

System	WER	(# Errors)
NICT	**13.5**	**(5,734)**
KIT	14.4	(6,115)
MITLL-AFRL	15.9	(6,788)
RWTH	16.0	(6,827)
NAIST	16.2	(6,897)
UEDIN	22.1	(9,413)
FBK	23.2	(9,899)
PRKE-IOIT	27.2	(11,578)

TED : ASR German (ASR$_{DE}$)

System	WER	(# Errors)
RWTH	**25.2**	**(4,845)**
KIT	25.7	(4,932)
FBK	37.5	(7,199)
UEDIN	37.8	(7,250)

TED : SLT English-French (SLT$_{EnFr}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
KIT	**26.81**	**55.08**	**27.53**	**54.06**
RWTH	25.62	57.21	26.41	56.09
UEDIN	22.45	61.34	23.30	60.06
MSR-FBK	22.42	63.69	23.72	62.20

TED : SLT English-German (SLT$_{EnDe}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
KIT	**18.05**	**64.46**	**18.66**	**63.22**
RWTH	17.27	66.33	17.88	65.09

TED : SLT German-English (SLT$_{DeEn}$)

System	*Ref. with disfluencies*				Ref. without disfluencies			
	case sensitive		case insensitive		*case sensitive*		case insensitive	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
KIT	**19.34**	**62.27**	**19.80**	**61.34**	**19.54**	**62.74**	**20.01**	**61.80**
UEDIN	14.92	68.12	15.39	67.28	15.03	68.70	15.52	67.86

TED : SLT English-Arabic (SLT$_{EnAr}$)

System	BLEU	TER
QCRI	**10.33**	**73.72**

TED : SLT English-Chinese (MT$_{EnZh}$)

System	*character-based*		*word-based*	
	BLEU	TER	BLEU	TER
KIT	**16.91**	**74.07**	**9.20**	**80.63**

TED : MT English-French (MT$_{EnFr}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
EU-BRIDGE	**38.86**	**42.96**	**39.74**	**42.02**
KIT	38.63	43.20	39.60	42.11
UEDIN	38.45	43.96	39.39	42.91
FBK	37.69	44.13	38.46	43.23
RWTH	37.67	44.00	38.49	43.04
PRKE-IOIT	37.59	45.07	38.39	44.15
MITLL-AFRL	37.05	45.36	38.27	44.10
BASELINE	31.94	48.59	32.56	47.75

TED : MT English-German (MT$_{EnDe}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
KIT	**25.71**	**54.46**	**26.47**	**53.34**
RWTH	24.74	55.52	25.41	54.42
NTT-NAIST	24.60	54.86	25.79	53.37
UEDIN	24.00	55.94	24.68	54.87
POSTECH	22.43	57.57	23.00	56.58
BASELINE	19.58	59.81	20.14	58.84

TED : MT German-English (SLT$_{DeEn}$)

System	*Ref. with disfluencies* *case sensitive* BLEU	*Ref. with disfluencies* *case sensitive* TER	*Ref. with disfluencies* case insensitive BLEU	*Ref. with disfluencies* case insensitive TER	Ref. without disfluencies *case sensitive* BLEU	Ref. without disfluencies *case sensitive* TER	Ref. without disfluencies case insensitive BLEU	Ref. without disfluencies case insensitive TER
KIT	**26.48**	57.52	**27.11**	56.60	**26.57**	58.31	**27.16**	57.41
EU-BRIDGE	26.33	**56.70**	26.91	**55.78**	**26.57**	**57.29**	27.14	**56.38**
NTT-NAIST	25.69	60.96	26.29	60.06	25.83	60.75	26.45	59.82
UEDIN	25.54	59.99	26.12	59.07	25.35	60.98	25.87	60.08
RWTH	25.32	59.67	25.94	58.67	25.27	60.46	25.86	59.51
HDU	22.91	59.65	23.94	58.35	23.06	60.38	24.07	59.11
POSTECH	21.26	67.61	21.74	66.72	21.17	68.91	21.65	68.04
BASELINE	19.25	65.03	19.79	64.19	19.07	65.94	19.55	65.11

TED : MT English-Arabic (MT$_{EnAr}$)

System	BLEU	TER
QCRI	**15.78**	**65.43**
KIT	15.51	65.64
BASELINE	12.12	68.73
UEDIN	11.49	70.58

TED : MT Arabic-English (MT$_{ArEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
QCRI	**30.49**	51.37	**31.21**	50.37
RWTH	29.95	**50.61**	31.07	**49.44**
MITLL-AFRL	26.64	55.17	27.54	54.05
UEDIN	26.29	56.69	26.92	55.70
BASELINE	22.71	59.02	23.52	57.94

TED : MT English-Spanish (MT$_{EnEs}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**34.74**	**45.75**	**35.42**	**44.79**
BASELINE	29.01	50.99	29.57	50.08

TED : MT Spanish-English (MT$_{EsEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**39.12**	**41.36**	**39.74**	**40.59**
BASELINE	33.18	45.58	33.68	45.00

TED : MT English-Farsi (MT$_{EnFa}$)

System	BLEU	TER
FBK	**10.12**	**71.58**
UEDIN	9.49	72.92
BASELINE	8.94	72.74

TED : MT Farsi-English (MT$_{FaEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
MITLL-AFRL	**16.03**	**78.82**	**16.51**	**77.84**
UEDIN	15.10	88.06	15.42	87.20
FBK	14.47	85.84	14.86	84.87
BASELINE	12.17	88.88	12.56	87.84

TED : MT English-Italian (MT$_{EnIt}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**29.17**	**50.84**	**29.90**	**49.87**
BASELINE	26.59	52.75	27.16	51.88

TED : MT Italian-English (MT$_{ItEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**34.89**	**47.50**	**35.55**	**46.64**
BASELINE	30.82	50.35	31.30	49.63

TED : MT English-Dutch (MT$_{EnNl}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**25.52**	**55.92**	**26.49**	**54.31**
BASELINE	22.82	57.66	23.54	56.33

TED : MT Dutch-English (MT$_{NlEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**32.73**	**51.32**	**33.74**	**49.93**
BASELINE	28.00	54.49	28.94	53.08

TED : MT English-Polish (MT$_{EnPl}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
PJIIT	**14.29**	**73.54**	**15.04**	**72.06**
UEDIN	11.51	77.66	12.03	76.48
BASELINE	10.31	76.19	10.79	75.05

TED : MT Polish-English (MT$_{PlEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
PJIIT	**22.60**	**62.56**	**23.54**	**61.12**
UEDIN	20.91	64.32	21.59	63.11
BASELINE	16.31	67.33	16.85	66.26

TED : MT English-Portuguese (MT$_{EnPt}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	**33.18**	**44.92**	**33.92**	**43.90**
BASELINE	29.65	46.85	30.18	46.06

TED : MT English-Romanian (MT$_{EnRo}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	**17.57**	**66.96**	**18.10**	**65.83**
BASELINE	16.18	68.29	16.70	67.16

TED : MT English-Russian(MT$_{EnRu}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	**16.14**	**70.28**	**16.15**	**69.12**
HDU	15.87	69.00	15.95	67.63
BASELINE	13.69	71.30	13.69	70.22

TED : MT English-Slovenian(MT$_{EnSl}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	**13.68**	**67.68**	**14.21**	**66.55**
RWTH	10.10	71.66	10.47	70.71
BASELINE	9.49	72.16	9.87	71.19

TED : MT English-Trukish(MT$_{EnTr}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
TUBITAK	**8.97**	**76.12**	**9.78**	**74.42**
UEDIN	6.76	82.32	7.24	81.09
BASELINE	6.62	79.96	6.94	78.80

TED : MT English-Chinese(MT$_{EnZh}$)

System	*character-based*		*word-based*	
	BLEU	TER	BLEU	TER
CASIA	**20.55**	**65.12**	**12.45**	**72.21**
KIT	19.83	69.75	11.47	76.72
HKUST	18.66	70.36	10.85	78.12
UEDIN	18.57	69.71	10.56	77.90
BASELINE	18.15	72.34	10.01	81.77

TED : MT Portuguese-English (MT$_{PtEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	**37.33**	**42.91**	**37.80**	**42.31**
BASELINE	35.80	42.93	36.14	42.44

TED : MT Romanian-English (MT$_{RoEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	**29.82**	**50.53**	**30.58**	**49.55**
BASELINE	24.85	54.21	25.46	53.23

TED : MT Russian-English (MT$_{RuEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
HDU	**23.78**	**59.51**	**25.00**	**58.04**
UEDIN	22.67	61.99	23.37	60.93
MITLL-AFRL	21.65	60.71	22.59	59.38
BASELINE	18.57	64.99	19.12	63.90

TED : MT Slovenian-English (MT$_{SlEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	**24.06**	**58.40**	**24.87**	**57.08**
RWTH	17.46	64.42	18.00	63.30
BASELINE	14.62	69.70	15.16	68.66

TED : MT Turkish-English (MT$_{TrEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
TUBITAK	**18.67**	**68.28**	**19.68**	**66.73**
UEDIN	14.87	74.19	15.63	72.85
BASELINE	12.24	75.90	12.89	74.79

TED : MT Chinese-English (MT$_{ZhEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
RWTH	**16.17**	**65.37**	**17.00**	**64.17**
UEDIN	15.26	69.73	15.91	68.61
MITLL-AFRL	14.85	68.99	15.53	67.85
CASIA	14.55	69.08	15.52	67.37
BASELINE	12.29	70.60	12.85	69.56
HKUST	9.58	74.82	10.17	73.75

A.2. Progress Testset (*tst2011*) and (*tst2012*)

· All the sentence IDs in the IWSLT 2011 testset were used to calculate the automatic scores for each run submission.
· ASR and MT systems are ordered according to the *WER* and *BLEU* metrics, respectively.
· For each task, the best score of each metric is marked with **boldface**.
· All automatic evaluation metric scores are given as percent figures (%).

TED : ASR English (ASR$_{EN}$)

tst2011

System	IWSLT 2011 WER	(# Errors)	IWSLT 2012 WER	(# Errors)	IWSLT 2013 WER	(# Errors)
FBK	16.2	(2,091)	15.4	(1,991)	13.6	(1,754)
KIT	15.0	(1,938)	12.0	(1,552)	9.3	(1,196)
MITLL-AFRL	13.5	(1,741)	11.1	(1,432)	10.6	(1,360)
NAIST	—		12.0	(1,553)	9.1	(1,172)
NICT	25.6	(3,301)	10.9	(1,401)	7.9	(1,016)
PRKE-IOIT	—		—		14.6	(1,883)
RWTH	—		13.4	(1,731)	10.2	(1,319)
UEDIN	—		—		10.2	(1,318)

tst2012

System	IWSLT 2012 WER	(# Errors)	IWSLT 2013 WER	(# Errors)
FBK	16.8	(3,227)	16.2	(3,090)
KIT	12.7	(2,435)	9.6	(1,834)
MITLL-AFRL	13.3	(2,565)	11.3	(1,360)
NAIST	12.4	(2,392)	10.0	(1,913)
NICT	12.1	(2,318)	8.6	(1,636)
PRKE-IOIT	—		16.2	(3,101)
RWTH	13.6	(2,621)	11.3	(2,166)
UEDIN	14.4	(2775)	11.6	(2,212)

TED : SLT English-French test 2012(SLT$_{EnFr}$)

System	*case sensitive* BLEU	TER	case insensitive BLEU	TER
KIT	32.21	48.58	32.86	47.65
MSR-FBK	29.92	53.30	31.03	52.10

TED : SLT English-French test2011(SLT$_{EnFr}$)

System	*case sensitive* BLEU	TER	case insensitive BLEU	TER
KIT	31.06	50.70	31.93	49.61
MSR-FBK	27.21	56.22	28.32	54.82

TED : MT English-French test 2012(MT$_{EnFr}$)

System	*case sensitive* BLEU	TER	*case insensitive* BLEU	TER
EU-BRIDGE	42.13	38.72	42.99	37.83
UEDIN	41.21	39.83	42.02	38.94
KIT	41.02	39.22	41.96	38.34
RWTH	40.06	39.95	40.79	39.11
PRKE-IOIT	39.94	41.52	40.64	40.75
MITLL-AFRL	39.76	41.47	40.97	40.31
FBK	39.51	40.56	40.11	39.80

TED : MT English-French test 2011(MT$_{EnFr}$)

System	*case sensitive* BLEU	TER	*case insensitive* BLEU	TER
EU-BRIDGE	40.71	40.56	41.55	39.72
UEDIN	40.61	40.97	41.48	40.08
MITLL-AFRL	39.35	42.18	40.62	41.08
RWTH	39.25	41.24	40.16	40.29
KIT	39.11	41.74	40.33	40.63
PRKE-IOIT	38.80	42.86	39.54	42.12
FBK	38.41	42.02	39.09	41.25

TED : MT English-German test 2012 (MT$_{EnDe}$)

System	*case sensitive* BLEU	TER	*case insensitive* BLEU	TER
KIT	23.24	56.17	24.00	55.02
NTT-NAIST	22.86	56.12	24.10	54.57
UEDIN	22.53	57.43	23.26	56.27
RWTH	22.32	57.11	23.04	55.91
POSTECH	20.43	59.14	21.02	58.05

TED : MT English-German test2011 (MT$_{EnDe}$)

System	*case sensitive* BLEU	TER	*case insensitive* BLEU	TER
UEDIN	27.13	50.97	27.75	50.09
KIT	26.29	50.67	26.97	49.76
NTT-NAIST	26.04	50.13	27.27	48.82
RWTH	25.86	51.56	26.58	50.52
POSTECH	23.48	53.71	24.06	52.89

TED : MT English-Arabic test 2012(MT$_{EnAr}$)

System	BLEU	TER
QCRI	15.54	65.57
KIT	15.07	66.46
UEDIN	12.37	69.79

TED : MT English-Arabic test 2011(MT$_{EnAr}$)

System	BLEU	TER
QCRI	15.54	69.19
KIT	14.59	70.60
UEDIN	11.90	72.60

TED : MT English-Spanish test 2012 (MT$_{EnEs}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	26.84	55.86	27.78	54.42

TED : MT English-Spanish test 2011 (MT$_{EnEs}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	33.17	47.77	34.02	46.59

TED : MT English-Farsi test 2012 (MT$_{EnFa}$)

System	BLEU	TER
FBK	10.94	72.66
UEDIN	10.24	74.24

TED : MT English-Farsi test 2011 (MT$_{EnFa}$)

System	BLEU	TER
FBK	12.55	70.06
UEDIN	12.29	71.73

TED : MT English-Italian test 2012(MT$_{EnIt}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	25.28	56.67	26.09	55.55

TED : MT English-Italian test 2011(MT$_{EnIt}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	24.40	57.35	25.15	56.30

TED : MT English-Dutch test 2012(MT$_{EnNl}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	26.66	53.21	27.74	51.62

TED : MT Arabic-English test 2012 (MT$_{ArEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
QCRI	30.26	49.55	31.13	48.51
RWTH	29.31	49.46	30.28	48.39
UEDIN	27.72	53.28	28.46	52.34
MITLL-AFRL	27.66	52.18	28.61	51.05

TED : MT Arabic-English test 2011 (MT$_{ArEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
QCRI	27.76	55.17	28.64	54.02
RWTH	27.34	54.41	28.52	53.05
MITLL-AFRL	25.66	57.60	26.58	56.32
UEDIN	25.58	58.91	26.25	57.89

TED : MT Spanish-English test 2011(MT$_{EsEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	30.78	48.65	31.67	47.48

TED : MT Spanish-English test 2012(MT$_{EsEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	37.09	43.45	38.08	42.21

TED : MT Farsi-English test 2012 (MT$_{FaEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	14.98	89.78	15.52	88.79
FBK	14.40	87.26	14.95	86.13

TED : MT Farsi-English test 2011 (MT$_{FaEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
MITLL-AFRL	20.04	62.76	20.90	61.55
UEDIN	19.15	67.64	19.80	66.49
FBK	18.85	66.38	19.48	65.20

TED : MT Italian-English test2012 (MT$_{ItEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	29.62	52.36	30.29	51.40

TED : MT Italian-English test2011 (MT$_{ItEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	30.24	51.81	31.04	50.81

TED : MT Dutch-English test2012 (MT$_{NlEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	33.02	47.96	34.46	46.19

TED : MT English-Dutch test 2011(MT$_{EnNl}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	30.33	47.48	31.54	45.92

TED : MT Dutch-English test2011 (MT$_{NlEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	36.02	45.55	37.36	43.75

TED : MT English-Polish test2012 (MT$_{EnPl}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
PJIIT	13.49	75.03	14.29	73.36
UEDIN	10.48	79.05	11.04	77.73

TED : MT Polish-English test2012 (MT$_{PlEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
PJIIT	19.77	65.34	20.75	63.79
UEDIN	18.51	66.75	19.39	65.33

TED : MT English-Polish test2011 (MT$_{EnPl}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
PJIIT	15.66	68.65	16.61	67.16
UEDIN	13.10	70.96	13.69	69.86

TED : MT Polish-English test2011 (MT$_{PlEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
PJIIT	23.29	60.99	24.37	59.36
UEDIN	21.69	62.73	22.57	61.24

TED : MT English-Portuguese test 2012(MT$_{EnPt}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	34.88	43.66	35.84	42.50

TED : MT Portuguese-English test 2012 (MT$_{PtEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	40.56	39.64	41.18	38.95

TED : MT English-Portuguese test 2011(MT$_{EnPt}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	33.59	44.41	34.40	43.37

TED : MT Portuguese-English test 2011 (MT$_{PtEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	39.02	41.24	39.66	40.43

TED : MT English-Romanian test 2012 (MT$_{EnRo}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	19.21	63.03	19.74	62.08

TED : MT Romainan-English test2012 (MT$_{RoEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	31.84	49.19	32.52	48.28

TED : MT English-Romanian test 2012 (MT$_{EnRo}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	23.19	56.60	23.77	55.72

TED : MT Romanian-English test2012 (MT$_{RoEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	36.05	43.99	36.92	42.90

TED : MT English-Russian test 2012(MT$_{EnRu}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
HDU	13.76	73.13	13.83	71.13
UEDIN	13.53	74.66	13.54	72.87

TED : MT Russian-English test 2012 (MT$_{RuEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	20.71	62.78	21.58	61.50
MITLL-AFRL	19.61	62.46	20.53	61.14
HDU	18.20	63.40	19.37	61.74

TED : MT English-Russian test 2011(MT$_{EnRu}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	15.93	67.63	15.94	66.45
HDU	15.53	67.43	15.61	65.79

TED : MT Russian-English test 2011 (MT$_{RuEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	22.13	61.24	22.82	60.05
MITLL-AFRL	21.49	60.10	22.41	58.74
HDU	20.16	61.72	21.30	60.22

TED : MT English-Slovenian test 2012 (MT$_{EnSl}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	12.35	70.12	12.88	69.05
RWTH	8.81	73.11	9.22	72.17

TED : MT Slovenian-English test2012 (MT$_{SlEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	21.20	61.54	22.03	60.27
RWTH	16.41	65.22	17.00	64.19

TED : MT English-Trukish test 2012 (MT$_{EnTr}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
TUBITAK	9.29	75.46	10.00	73.85
UEDIN	7.41	81.67	7.84	80.20

TED : MT English-Trukish test 2011 (MT$_{EnTr}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
TUBITAK	9.16	75.89	10.19	73.90
UEDIN	7.36	81.30	8.14	79.57

TED : MT English-Chinese test2012 (MT$_{EnZh}$)

System	*character-based*		*word-based*	
	BLEU	TER	BLEU	TER
CASIA	21.88	65.57	13.41	72.64
UEDIN	18.07	71.31	10.80	79.72
KIT	17.93	73.04	10.04	80.39

TED : MT English-Chinese test2011 (MT$_{EnZh}$)

System	*character-based*		*word-based*	
	BLEU	TER	BLEU	TER
CASIA	24.04	62.90	14.94	70.60
KIT	20.41	69.37	11.76	77.88
UEDIN	19.75	68.51	11.54	78.20

TED : MT Turkish-English test 2012 (MT$_{TrEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
TUBITAK	18.93	67.03	19.84	65.49
UEDIN	15.00	72.58	15.77	71.38

TED : MT Turkish-English test 2011 (MT$_{TrEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
TUBITAK	18.63	67.60	19.61	65.99
UEDIN	15.02	73.90	15.89	72.53

TED : MT Chinese-English test 2012(MT$_{ZhEn}$)

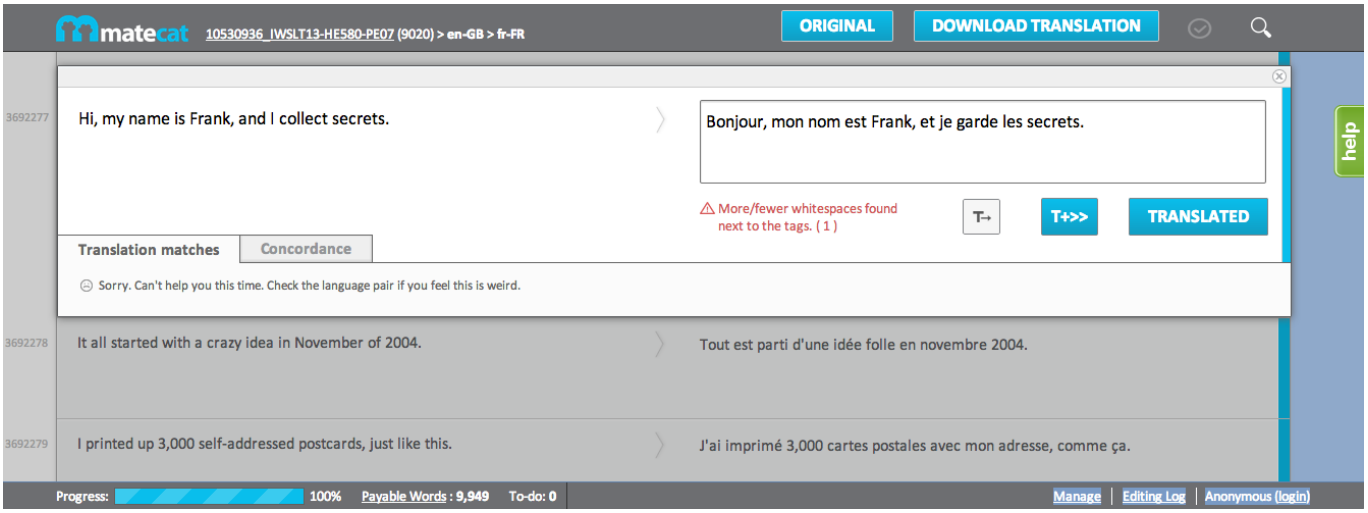
System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
RWTH	14.62	65.73	15.64	64.17
UEDIN	14.19	68.93	15.02	67.54
MITLL-AFRL	14.05	68.26	14.92	66.85
CASIA	12.36	68.76	13.52	66.98

TED : MT Chinese-English test 2011(MT$_{ZhEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
RWTH	16.61	63.37	17.57	61.96
UEDIN	16.10	65.45	16.82	64.18
MITLL-AFRL	15.92	65.68	16.82	64.40
CASIA	14.40	65.60	15.32	64.01

Appendix B. Human Evaluation

Interface used for the bilingual post-editing task



Post-editing instructions given to professional translators

In this task you are presented with automatic translations of TED Talks captions.

You are asked to post-edit the given automatic translation by applying the minimal edits required to transform the system output into a fluent sentence with the same meaning as the source sentence.

While post-editing, remember that the post-edited sentence is to be intended as a transcription of spoken language. Note also that the focus is the correctness of the single sentence within the given context, NOT the consistency of a group of sentences. Hence, surrounding segments should be used to understand the context but NOT to enforce consistency on the use of terms. In particular, different but correct translations of terms across segments should not be corrected.

Examples:

Source: This next one takes a little explanation before I share it with you.
Automatic translation: ...avant que je partage avec vous.
Post-editing 1: ...avant de le partager avec vous.
Post-editing 2: ...avant que je le partage avec vous. (preferred - minimal editing and acceptable in spoken language)

Source: And the table form is important.
Automatic translation: Et la forme de la table est importante.
Post-editing 1: La forme de la table est également importante.
Post-editing 2: Et la forme de la table est importante. (preferred - no editing - slightly less fluent but better fitting the source speech transcription)

Source: Everyone who knew me before 9/11 believes...
Automatic translation: ...avant le 11/9...
Post-editing 1: ...avant le 11 septembre...
Post-editing 2: ...avant le 11/9... (preferred - no editing - better fitting the source)

Human Semantic MT Evaluation with HMEANT for IWSLT 2013

Chi-kiu Lo Dekai Wu

HKUST
Human Language Technology Center
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
{jackielo|dekai}@cs.ust.hk

Abstract

We present the results of large-scale human semantic MT evaluation with HMEANT on the IWSLT 2013 German-English MT and SLT tracks and show that HMEANT evaluates the performance of the MT systems differently compared to BLEU and TER. Together with the references, all the translations are annotated by annotators who are native English speakers in both semantic role labeling stage and role filler alignment stage of HMEANT. We obtain high inter-annotator agreement and low annotation time costs which indicate that it is feasible to run a large-scale human semantic MT evaluation campaign using HMEANT. Our results also show that HMEANT is a robust and reliable semantic MT evaluation metric for running large-scale evaluation campaigns as it is inexpensive and simple while maintaining the semantic representational transparency to provide a perspective which is different from BLEU and TER in order to understand the performance of the state-of-the-art MT systems.

1. Introduction

This paper presents the results from the human semantic MT evaluation with HMEANT on the IWSLT 2013 German-English MT and SLT tracks which show that HMEANT provides a perspective which is different from BLEU and TER in evaluating the performance of the MT systems. The IWSLT evaluation campaign has offered a variety of speech translation tasks over the past decade but none of them included evaluation of system performance using a semantic MT evaluation metric because of the inherent cost in evaluation in terms of both the (a) amount of time, and (b) the level of expertise needed by the human annotators. We choose HMEANT as a way around these challenges given substantial empirical evidence [1, 2] that HMEANT is a inexpensive, simple, and representationally transparent semantic MT evaluation metric that correlates with human translation adequacy judgements more highly than HTER [3] and other automatic MT evaluation metrics, such as BLEU [4], NIST [5], METEOR [6], PER [7], CDER [8], WER [9], and TER [3].

Although fast and inexpensive lexical n-gram based objective functions like BLEU have driven MT system development over the past decade, these metrics do not enforce translation utility adequately and often fail to preserve meaning [10, 11]. We believe that the system development should also be driven by semantic MT evaluation metrics which focus on getting the meaning right. Recent results [12, 13, 14] which indicate that more adequate translations are produced by tuning MT systems using the semantic evaluation metric MEANT, support us.

In this paper, we present the results of one of the largest semantic MT evaluations to date, in terms of both the number of systems and the number of translations evaluated, using HMEANT as the evaluation metric. The aims of this evaluation campaign are two-fold: (1) to demonstrate feasibility of running a large-scale semantic MT evaluation campaign using humans, and (2) to provide fine-grained statistics over a large number of systems that enable a fair comparison of semantic human MT evaluation metrics and other automatic metrics. While the former goal helps realize a practical semantically driven human MT evaluation metric in the place of expensive human MT evaluation metrics such as HTER or simple translation ranking which does not adequately reflect translation utility. The latter goal not only provides useful insights into the differences between metrics gauging semantic similarity and surface based metrics, but also quantifies the robustness

of HMEANT as an MT evaluation metric.

In the rest of the paper, we discuss the details of the evaluation campaign and provide results on the inter-annotator agreement on the tasks of semantic role annotation and alignment. We also provide an analysis of the time taken for annotation and the alignment of the semantic roles. We also report the results of different participating systems according to the criterion of our semantic evaluation metric HMEANT and its automatic variant, MEANT [15].

2. Participating tracks and systems

To perform a full-scale semantic MT evaluation, all the systems which participated in IWSLT 2013 German-English MT and SLT tracks were evaluated. There were 17 systems participating in the MT track and 3 systems participating in the SLT track.

The evaluation set consists of 136 sentences randomly drawn from the test set *(tst2013)*, which represents around 10% of the entire test set. The systems from the MT track are evaluated against the reference without disfluencies while the systems from the SLT track are evaluated against the reference with disfluencies. The details description of the tracks, the original test set and the participating systems can be found in the overview paper of IWSLT 2013 [16].

This is the largest scale semantic MT evaluation using HMEANT to date, in terms of both the number of systems and the number of translations evaluated.

3. HMEANT

HMEANT is the weighted f-score over matching semantic roles between the reference and the MT output, where the labeling and alignment of frames and role fillers is performed manually by minimally trained annotators. HMEANT, which can be driven by low-cost monolinguals of the output language, not only outperforms the commonly used automatic MT evaluation metrics, such as, BLEU, NIST, METEOR, WER, CDER and TER, but also outperforms HTER in correlating with human adequacy judgment at much lower labor cost.

HMEANT is computed as follows:

1. Human annotators annotate the shallow semantic structures of both the reference and the MT output (Figure 1 shows examples of human shallow semantic parses on both reference and MT output.)
2. Human judges align the semantic frames between the references and the MT output by judging the correctness of the predicates.
3. For each pair of aligned semantic frames,
   (a) Human judges determine the translation correctness of the semantic role fillers.
   (b) Human judges align the semantic role fillers between the reference and the MT output according to the correctness of the semantic role fillers.
4. Compute the weighted f-score over the matching role labels of these aligned predicates and role fillers according to the mathematical definitions in the following.

$$
\begin{aligned}
M_{i,j} &\equiv \text{total \# ARG j of aligned frame i in MT} \\
R_{i,j} &\equiv \text{total \# ARG j of aligned frame i in REF} \\
C_{i,j} &\equiv \text{\# correct ARG j of aligned frame i} \\
P_{i,j} &\equiv \text{\# partially correct ARG j of aligned frame i} \\
w_{\text{pred}} &\equiv \text{weight of similarity of predicates} \\
w_j &\equiv \text{weight of similarity of ARG j} \\
w_{\text{partial}} &\equiv \text{weight of the partially correct translated ARG}
\end{aligned}
$$

$$
\begin{aligned}
m_i &\equiv \frac{\text{\#tokens filled in aligned frame i of MT}}{\text{total \#tokens in MT}} \\
r_i &\equiv \frac{\text{\#tokens filled in aligned frame i of REF}}{\text{total \#tokens in REF}}
\end{aligned}
$$

$$
\text{precision} = \frac{\sum_i m_i \frac{w_{\text{pred}} + \sum_j w_j \, (C_{i,j} + w_{\text{partial}} P_{i,j})}{w_{\text{pred}} + \sum_j w_j M_{i,j}}}{\sum_i m_i}
$$

$$
\text{recall} = \frac{\sum_i r_i \frac{w_{\text{pred}} S_{i,\text{pred}} + \sum_j w_j (C_{i,j} + w_{\text{partial}} P_{i,j})}{w_{\text{pred}} + \sum_j w_j R_{i,j}}}{\sum_i r_i}
$$

where m_i and r_i are the weights for frame, i, in the MT/REF respectively. These weights estimate the degree of contribution of each frame to the overall meaning of the sentence. $M_{\text{i,j}}$ and $R_{\text{i,j}}$ are the total counts of argument of type j in frame i in the MT and REF respectively. $C_{i,j}$ and $P_{i,j}$ are the count of the correctly and partially correct translated argument j in frame i in the MT output. The weights w_{pred} and w_j are the weights of the predicates and role fillers of the arguments of type j between the reference translations and the MT output.

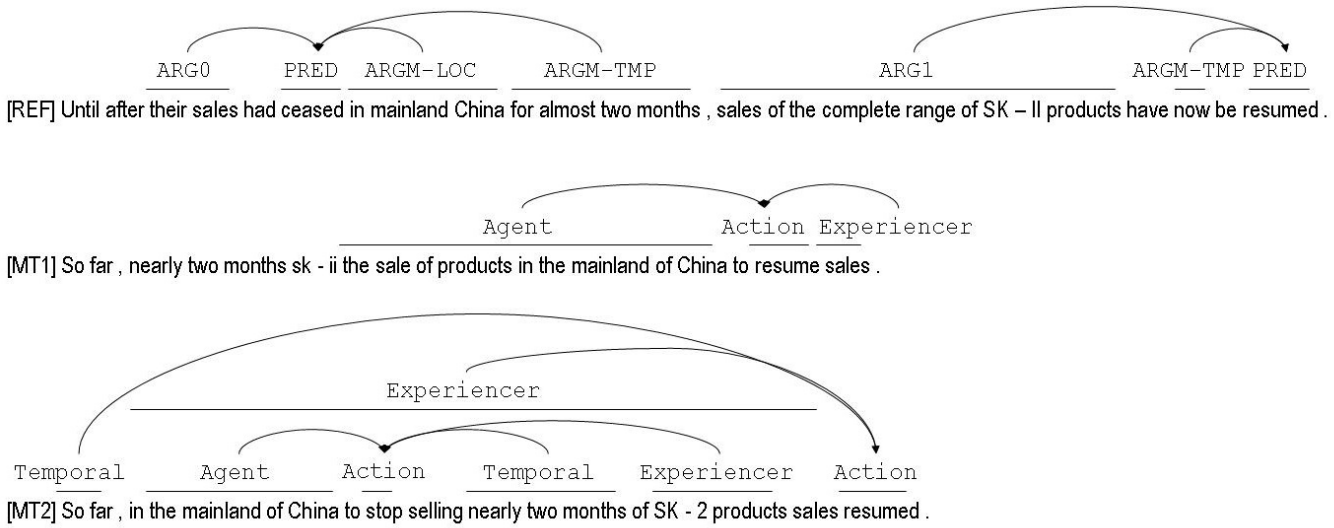


Figure 1: Examples of human semantic role labeling. There are no semantic frames for MT3 since there is no predicate.

Table 1: Example of SRL annotation for the MT2 output from Figure 1 along with the human judgements of translation correctness for each argument. *Notice that although the decision made by the human judge for "in mainland China" in the reference translation and "the mainland of China" in MT2 is "correct", nevertheless the HMEANT computation will not count this as a match since their role labels do not match.

REF roles	**REF**	**MT2 roles**	**MT2**	**decision**
PRED	ceased	Action	stop	match
ARG0	their sale	—	—	incorrect
ARGM-LOC	in mainland China	Agent	the mainland of China	correct*
ARGM-TMP	for almost two months	Temporal	nearly two months	correct
—	—	Experiencer	SK - 2 products	incorrect
PRED	resumed	Action	resume	match
ARG0	sales of complete range of SK - II products	Experiencer	in the mainland of China to stop selling nearly two months of SK - 2 products sales	incorrect
ARGM-TMP	Until after , their sales had ceased in mainland China for almost two months	Temporal	So far	partial
ARGM-TMP	now	—	—	incorrect

The weight w_{partial} is the weight of the partially correct translated arguments. There is a total of 12 weights for the set of semantic role labels in MEANT as defined in [17] and a weight for the partially correct translated arguments. These weights can be determined using supervised estimation via a simple grid search to optimize the correlation with human adequacy judgments [1] or like UMEANT, estimated in an unsupervised manner using relative frequency of each semantic role label in the reference translations. U(H)MEANT can thus be used when when human judgments on adequacy of the development set are unavailable [18].

Figure 1 shows examples of human judges' decisions for semantic frame annotation on the reference and the MT output. Table 1 shows examples of the human judges' decisions for semantic frame alignment

Table 2: Inter-annotator agreement for the human semantic role labeling task.

	reference	MT output
IAA	80.86%	72.69%

and translation correctness for each semantic role for the "MT2" output in Figure 1.

4. Human annotation

HMEANT consists of two human annotation steps: (1) human semantic role labeling, which labels semantic frames within the translations and (2) human role filler alignment that determines the correctness of the translation according to the captured meaning structures. We run the human annotation using HKUST's efficient and user friendly HMEANT web-based user interface workflow [19].

4.1. Semantic role labeling

Human semantic role labeling was carried out on the references and all the submitted German-English systems in the MT track and the SLT track to capture the meaning of the translation into the "**who** *did* **what** to **whom**, **when**, **where**, **why** and **how**" structure.

4.1.1. Task description and setup

As opposed to HTER which is driven by professional bilingual translators, the semantic role labeling task in HMEANT is driven by monolinguals with minimal training of 15 minutes. To increase the robustness of the human semantic role labeling, we increased the training time for the annotators from 15 minutes to 20 minutes. The additional 5 minutes contribute to showing more annotated examples that demonstrate how to annotate the ungrammatical MT output.

Each system was annotated by two annotators who are native English speakers to support estimation of the annotation reliability. In addition, each annotator labeled the sentences from the evaluation set only once to prevent them from getting extra out-of-context information in understanding the meaning of the translation.

4.1.2. Inter-annotator agreement and time efficiency

Table 2 shows that the IAA is over 80% for labeling the semantic roles manually in the reference translation and over 72% in the MT output. The high IAA shows that the human semantic role labeling is robust and reliable.

Previous work shows that it takes the minimally trained annotators approximately 1.5 minutes to finish labeling the semantic roles of one translation output. In this evaluation, the average time needed to label the semantic roles for one translation output is significantly decreased to 50 seconds due to the fact that the semantic structures of the TED talk sentences are simpler than formal newswire text.

Table 3: Inter-annotator agreement for the human role filler alignment task.

	alignment
IAA	63.23%

4.2. Semantic role filler alignment

Human semantic role filler alignment was carried out between the references and all the submitted German-English systems in the MT track and the SLT track to determine the translation correctness according to the captured semantic structures in the previous human semantic role labeling step.

4.2.1. Task description and setup

Similar to human semantic role labeling task, we increased the training time for the native English speaking annotators by 5 minutes for showing more examples that demonstrate how to align the ungrammatical MT output to the reference.

To support the reliability analysis of the evaluation, each system was annotated by two annotators. Since the annotators are constrained to determine the phrasal translation correctness of the labeled role fillers only, it is less likely that they could be contaminated by the out-of-context information acquired due to seeing translations of the same sentence more than once. Therefore, a single annotator allowed to align translations of the same sentence from different systems.

4.2.2. Inter-annotator agreement and time efficiency

Table 3 shows that the IAA is over 63% for aligning the semantic role fillers between the reference and the MT output. The high IAA shows that the human semantic role filler alignment task is robust and reliable.

Similar to the human semantic role labeling task,

Table 4: HMEANT, MEANT, BLEU and TER scores of all the systems participating in the IWSLT 2013 German-English MT track on the evaluation set randomly drawn from tst2013 where the BLEU and TER scores are the results of the official case insensitive, without disfluencies evaluation[16]. Italicized scores indicate systems that are ranked differently from HMEANT by the corresponding metrics.

system	HMEANT	MEANT	BLEU	TER
KIT.primary	**56.55**	48.90	**27.16**	57.41
KIT.contrastive1	55.99	48.36		
EU-BRIDGE.primary	55.89	***48.97***	27.14	***56.38***
EU-BRIDGE.contrastive1	55.62	47.28		
KIT.contrastive2	55.11	*46.87*		
UEDIN.primary	54.84	47.13	25.87	*60.08*
RWTH.primary	54.63	46.51	25.86	59.51
RWTH.contrastive	54.46	46.44		
NTT-NAIST.primary	54.01	46.02	*26.45*	59.82
HDU.primary	53.99	45.99	24.07	*59.11*
HDU.contrastive2	52.47	45.37		
HDU.contrastive1	51.54	44.96		
NTT-NAIST.contrastive1	51.35	44.09		
NTT-NAIST.contrastive2	50.29	42.78		
NTT-NAIST.contrastive3	49.74	42.04		
Baseline	49.12	41.91	19.55	65.11
KLE.primary	44.53	*43.91*	*21.65*	68.04

Table 5: HMEANT and MEANT scores of all the systems participating in the IWSLT 2013 German-English SLT track on the evaluation set randomly draw from tst2013 where the BLEU and TER scores are the results of the official case insensitive, with disfluencies evaluation[16]. Italicized scores indicate systems that are ranked differently from HMEANT by the corresponding metrics.

system	HMEANT	MEANT	BLEU	TER
KIT.primary	**45.96**	**37.54**	**19.80**	**61.34**
UEDIN.primary	40.05	35.39	15.39	67.28
UEDIN.contrastive1	37.18	33.55		

in this evaluation the average time taken by minimally trained annotators to align the semantic roles between the reference and the MT output significantly decreases from 1.5 minutes to 42 seconds because the semantic structures of the TED talk sentences are simpler compared to formal newswire text. HMEANT scores are calculated by averaging the scores obtained from the two different annotations in each of the annotation task.

5. Results

From the results one can observe how HMEANT and MEANT provide different rankings compared to BLEU and TER. All four metrics HMEANT, MEANT, BLEU and TER rate KIT primary and EU-BRIDGE primary systems as closely tied in the first place according to the numbers in Table 4. On the other hand, while BLEU claims that NTT-NAIST primary system significantly outperforms UEDIN, RWTH, and HDU, both HMEANT and MEANT indicate that all four teams in the middle actually achieved comparable results. Surprisingly, HDU which is ranked the best system according to TER is ranked worst according to BLEU. These differences in the ranking of different systems between HMEANT, BLEU and TER indicates that HMEANT does offer a different perspective compared to BLEU and TER. Further, the evidence for the high correlation of HMEANT with human adequacy judgement makes HMEANT an ideal candidate for human semantic MT evaluation. Table 5 reports the scores of all four met-

rics for the three systems in the SLT track. Between KIT.primary and UEDIN.primary, all the metrics agree that KIT is better by a wide margin.

From Table 4, we can also notice that the HMEANT score of KLE.primary system is significantly smaller than the other systems. This is because the annotators failed to understand the translation output of the KLE.primary system due to excessive amounts of punctuations and symbols in the translations. Unlike BLEU score which is better than the baseline, this dearth of adequacy is appropriately represented by a sharp decrease in the HMEANT score (compared to the baseline) indicating that HMEANT reflects the translation adequacy when traditional evaluation metrics like BLEU fail to do so.

6. Conclusion

We presented the results of human semantic MT evaluation with HMEANT on the IWSLT 2013 German-English MT and SLT tracks. We also showed that rankings provided by HMEANT are different compared to BLEU and TER thereby offering a different perspective on evaluating MT system performance. The empirical evidence for HMEANT's high correlation with human judgement on translation adequacy, its semantic motivation and representational transparency makes HMEANT a viable human semantic MT evaluation metric. Further, the high inter-annotator agreement and low annotation time cost as demonstrated in this evaluation indicate that HMEANT is robust, reliable and efficient to run a large scale human semantic MT evaluation. Given our results, we believe that it would be essential to include HMEANT in evaluation campaigns so as to provide a different semantically motivated view of the state-of-the-art MT system performance to the research community.

7. Acknowledgments

This material is based upon work supported in part by the European Union under the FP7 grant agreement no. 287658; by the Defense Advanced Research Projects Agency (DARPA) under BOLT contract no. HR0011-12-C-0016, and GALE contract nos. HR0011-06-C-0022 and HR0011-06-C-0023; and by the Hong Kong Research Grants Council (RGC) research grants GRF620811, GRF621008, and GRF612806. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the EU, or RGC. We are grateful to all the annotators for running the evaluation in the short period of time. Thanks to Karteek Addanki for assistance with editing the paper.

8. References

[1] Chi-kiu Lo and Dekai Wu, "MEANT: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles," in *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT 2011)*, 2011.

[2] ——, "Structured vs. flat semantic role representations for machine translation evaluation," in *Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-5)*, 2011.

[3] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul, "A study of translation edit rate with targeted human annotation," in *7th Biennial Conference Association for Machine Translation in the Americas (AMTA 2006)*, Cambridge, Massachusetts, August 2006, pp. 223–231.

[4] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "BLEU: a method for automatic evaluation of machine translation," in *40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, Pennsylvania, July 2002, pp. 311–318.

[5] George Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *The second international conference on Human Language Technology Research (HLT '02)*, San Diego, California, 2002, pp. 138–145.

[6] Satanjeev Banerjee and Alon Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, Michigan, June 2005, pp. 65–72. [Online]. Available: http://www.aclweb.org/anthology/W/W05/W05-0909

[7] Christoph Tillmann, Stephan Vogel, Hermann Ney, Arkaitz Zubiaga, and Hassan Sawaf, "Accelerated DP based search for statistical translation," in *Fifth European Conference on Speech Communication and Technology (EUROSPEECH 1997)*, 1997.

[8] Gregor Leusch, Nicola Ueffing, and Hermann Ney, "CDer: Efficient MT evaluation using block movements," in *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 2006.

[9] Sonja Nießen, Franz Josef Och, Gregor Leusch, and Hermann Ney, "A evaluation tool for machine translation: Fast evaluation for MT research," in *The Second International Conference on Language Resources and Evaluation (LREC 2000)*, 2000.

[10] Chris Callison-Burch, Miles Osborne, and Philipp Koehn, "Re-evaluating the role of BLEU in machine translation research," in *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 2006, pp. 249–256.

[11] Philipp Koehn and Christof Monz, "Manual and automatic evaluation of machine translation between european languages," in *Workshop on Statistical Machine Translation (WMT-06)*, 2006, pp. 102–121.

[12] Chi-kiu Lo, Karteek Addanki, Markus Saers, and Dekai Wu, "Improving machine translation by training against an automatic semantic frame based evaluation metric," in *51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, 2013.

[13] Chi-kiu Lo and Dekai Wu, "Can informal genres be better translated by tuning on automatic semantic metrics?" in *14th Machine Translation Summit (MT Summit XIV)*, 2013.

[14] Chi-kiu Lo, Meriem Beloucif, and Dekai Wu, "Improving machine translation into Chinese by tuning against Chinese MEANT," in *International Workshop on Spoken Language Translation (IWSLT 2013)*, 2013.

[15] Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu, "Fully automatic semantic MT evaluation," in *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012.

[16] Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico, "Report on the 10th IWSLT evaluation campaign," in *International Workshop on Spoken Language Translation (IWSLT 2013)*, 2013.

[17] Chi-kiu Lo and Dekai Wu, "SMT vs. AI redux: How semantic frames evaluate MT more accurately," in *Twenty-second International Joint Conference on Artificial Intelligence (IJCAI-11)*, 2011.

[18] ——, "Unsupervised vs. supervised weight estimation for semantic MT evaluation metrics," in *Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-6)*, 2012.

[19] ——, "A radically simple, effective annotation and alignment methodology for semantic frame based SMT and MT evaluation," in *International Workshop on Using Linguistic Information for Hybrid Machine Translation (LIHMT-2011)*, 2011.

Edinburgh SLT and MT System Description for the IWSLT 2013 Evaluation

Alexandra Birch, Nadir Durrani, Philipp Koehn

School of Informatics
University of Edinburgh
Scotland, United Kingdom
a.birch@ed.ac.uk {dnadir,pkoehn}@inf.ed.ac.uk

Abstract

This paper gives a description of the University of Edinburgh's (UEDIN) systems for IWSLT 2013. We participated in all the MT tracks and the German-to-English and English-to-French SLT tracks. Our SLT submissions experimented with including ASR uncertainty into the decoding process via confusion networks, and looked at different ways of punctuating ASR output. Our MT submissions are mainly based on a system used in the recent evaluation campaign at the Workshop on Statistical Machine Translation [1]. We additionally explored the use of generalized representations (Brown clusters, POS and morphological tags) translating out of English into European languages.

1. Spoken Language Translation

We submit two systems to the Spoken Language Translation track: English-French and German-English. These systems were built to take maximum advantage of Edinburgh's English [2] and German [3] 2013 IWSLT speech recognition systems.

We explored different strategies for minimizing the mismatch between unpunctuated ASR output and SMT models, which are typically trained on punctuated text. We wanted to examine whether it was better to infer punctuation in the target during the translation process, or whether it was better to resolve ambiguity in the source first, by punctuating ASR output before translation. Previous work [4] has shown that it is helpful to punctuate ASR before translating, especially when using a strong punctuation model.

We also investigate how best to use the uncertainty in the ASR output. Confusion networks have been used successfully in speech translation [5]. They were proposed as a way to simplify ASR word graphs [6] as each path from the start node to the end node goes through all the other nodes. We compared using confusion networks from our speech systems to 1-best input into the machine translation models.

1.1. ASR systems

The english ASR system combines tandem and hybrid deep neural network based acoustic models, and applied adaptation to each speaker in the test set. N-best lists produced with an n-gram language model are rescored with a recurrent neural network language model to produce the final results. For more details see [2].

The German ASR lattices were generated using the KALDI speech recognition toolkit [7]. A hybrid deep neural network architecture was trained, in which a DNN with six hidden layers, containing 2048 nodes each, takes 39-dimensional speaker-adapted LDA-MLLT feature vectors as input to generate posterior probabilities over the 3000 context-dependent states of a HMM. Language modelling was done with a 4-gram LM which was trained on approximately 30 million words, selected from a text corpus of 994 million words, according to maximal cross-entropy with the TED domain. The lexicon was restricted to 300,000 words, striking a balance between adequate word coverage and low perplexity on the TED domain. The lattices were first generated with a heavily pruned version of this LM, and then rescored with the full model. For details, see [3].

1.2. Experimental design

We trained a phrase-based model using Moses [8] on the parallel corpora described in Table 1. These are large parallel corpora, with only TED talks [9] consisting of in-domain data. Europarl v7 [10], News Commentary corpus and Multi United Nations corpus [11], Gigaword corpus (French Gigaword Second Edition, English Gigaword Fifth Edition) and Common Crawl [12] consist of parallel data which contain some noise, and a large number of examples which are likely irrelevant for the target TED domain. We therefore used a domain filtering technique [13] which was applied successfully in last year's Edinburgh submission [14]. This uses bilingual cross-entropy difference to select sentence pairs that are similar to the in-domain data and dissimilar to the out-of-domain data. For French-English we retained 10% of the out-of-domain data, and for German-English, which has less out-of-domain data, we retain 20%.

To optimize the translation model we used a modified version of the MIRA implementation in Moses as described in [15]. The language model used is a 5-gram language model, trained with SRILM [16], and applies Kesner-Ney smoothing. The final model is a linear interpolation of language models trained separately on the corpora listed in the

Parallel Corpora	en-fr	de-en
TED(In Domain)	2.7/2.4	2.6/2.7
Europarl v7	52.8/58.2	48.7/42.5
News Commentary v7	3.4/3.9	4.0/3.9
Common Crawl	78.1/86.4	49.5/53.1
Multi UN	318.4/366.8	4.4/4.6
10^9	562.1/667.3	-
Monolingual Corpora	fr	en
TED(In Domain)	3.1	2.8
Europarl v7	61.5	60.5
News Commentary v7	4.0	3.9
Common Crawl	91.4	59.8
Multi UN	426.8	-
10^9	811.4	-

Table 1: Word counts (in millions) for corpora used to train translation and language models.

	tst2010
In+100%Out	30.8
In+10%Out	31.6 (+0.8)
In+10%Out, Strip Punc	28.4 (-3.2)

Table 2: Cased BLEU results for English-French baseline models when tuned and tested on gold transcriptions.

bottom half of Table 1. The interpolation is done to optimize entropy on the development set. For the German-English systems we applied compound splitting [17] and syntactic pre-ordering [18] on the German source side.

1.3. Baseline

In these experiments we establish what is the best baseline model to use for further spoken language translation experiments. Here we tune and test on transcribed TED talks. For both French-English and German-English the tuning set is their respective IWSLT dev2010 set, and the test set is their respective IWSLT tst2010 set.

Table 2 presents the results of the English-French baseline experiments. We can see that filtering the out-of-domain data not only reduced model size, but it increases performance by 0.8 BLEU points. We then wanted to test what effect the lack of punctuation has on performance, without the confounding factor of possible speech recognition errors. So we tested our filtered model with a test set for which punctuation on the source had been removed. In this paper, whenever punctuation is stripped we exclude full stops in acronyms such as "U.K." and quotes such as "we'll", as these occur in ASR output. We can see that performance is severely degraded by 3.2 BLEU points. This shows that punctuation alone accounts for a large part of the challenge in the speech translation task.

Table 3 shows the results of the German-English baseline experiments. We can see that filtering the out-of-domain data had a big increase on performance, 6.4 BLEU points. This means that out-of-domain data is either of poor quality or is badly mismatched with the test domain. For experiments with confusion networks, we would be unable to split and preorder the input. We therefore experimented with removing this preprocessing step. We can see that it has a big negative effect on the translation quality, losing 3.5 BLEU points. Although syntactic preordering of German input is very helpful for transcriptions, it is logical to suppose that applying it to ASR output with many errors would be less successful. We then experimented further, removing punctuation to reproduce the format of ASR input, and we lost a further 0.7 BLEU points.

	tst2010
In+100%Out	21.4
In+20%Out	27.8 (+6.4)
In+20%Out, No preord	24.3 (-3.5)
In+20%Out, No preord, Strip Punc	23.6 (-0.7)

Table 3: Cased BLEU results for German-English baseline models when tuned and tested on gold transcriptions.

1.4. Dealing with Uncertainty

In this section we explore the different ways that MT systems are able to use the uncertainty inherent in the ASR output, especially looking at punctuation insertion and confusion networks. We apply two models (with and without punctuation on the input) from the baseline experiments, the final two models in Table 2 and Table 3. The input to these experiments is the 1-best ASR output and confusion network ASR output from the Edinburgh ASR system submissions. For French-English the tuning set is dev2010 and the test set is tst2010. For German-English the tuning set is dev2012 and there is no test set, so results are reported for development data which is far from ideal.

The Kaldi and the HTK lattices were converted into standard lattice format and then into confusion networks or word meshes using the SRILM nbest-lattice tool. In speech recognition systems, high accuracy recognition is achieved by a multi-pass process which often use lattices as an intermediate representation. These lattices routinely contain redundant information which was generated due to small differences in timing. There could be, for instance, 10 different arcs emitting the same word with slightly different start times. This greatly increases the size and difficulty in translating the ASR output. We therefore apply a reduction step to the lattices [19], which reduced their average size by a factor of five. We set the number of iterations for reduction to 3. We also calculate the posterior probability of the arcs, pruning arcs with a variety of different thresholds, from 0.01 times the most likely candidate to 0.0001 times the most likely candidate. Finally we remove arcs which emit null.

	BLEU
Absolute 1-best	22.9
Absolute 1-best Punctuated	24.1 (+1.2)
Lattice 1-best	17.9 (-5.0)
CN prune p.t. 100	19.5 (+1.6)
CN prune p.t. 20	19.5 (+1.6)
CN prune p.t. 10	19.2 (+1.3)
CN prune p.t. 1	14.6 (-3.3)
CN prune p.t. 100 lattice 0.0001	19.3 (+1.4)
CN prune p.t. 100 lattice 0.001	19.3 (+1.4)
CN prune p.t. 100 lattice 0.01	19.4 (+1.5)

Table 4: Cased BLEU scores and decoding times in minutes for en-fr models when tuned and tested on ASR output.

We apply standard tokenization strategies to all languages. For confusion networks we need to split the arcs which carry a word which needs splitting. For instance an arc with the word "Europe's" becomse two arcs: "Europe" and "'s". We apply truecasing to all training and test data, including confusion networks. Truecasing models are trained on the tokenized parallel corpora. The most common case for a word is then applied to all text.

The punctuation SMT model is trained on monolingual data where the source side has had all punctuation stripped. This model is run in a monotone decoding mode so as to introduce as few changes as possible, limiting it as much as possible to just inserting punctuation.

The results for the extensive en-fr experiments are presented in Table 4. We first experimented with taking the absolute ASR 1-best output and using this for tuning and testing. We can see that it has a BLEU score of 22.9. We use this as the baseline result for comparison for the next results. We then compared this with our punctuated model. This model first passes the absolute 1-best through our SMT punctuation model. We can see that this improves results considerably, adding 1.2 points to the BLEU score. The absolute 1-best is the result of minimum Bayes risk decoding and system combination, where the lattices from the tandem and hybrid deep neural network based acoustic models are combined using ROVER. For our lattice and confusion network experiments however, we use the lattice output from the hybrid system. We lose some performance because not only do we miss out on the benefits of system combination, but we also do not benefit from a 4-gram language model and a final recurrent neural network language model rescoring step. In the English ASR paper [2], the absolute 1-best has a WER of 17.0, and the hybrid system has a WER of 18.6. We therefore include as our next system, the 1-best that we extract from the hybrid model's lattices using SRILM lattice-tool. The hybrid lattice 1-best has a BLEU score of 17.94, which is a drop of BLEU score of 5 points from the absolute 1-best. This is a surprisingly large negative impact considering that the WER of the hybrid system was only 1.6 points higher. Clearly the quality of the ASR system is of crucial importance to the final translation. We use the BLEU score of the hybrid lattice 1-best to compare the performance of the confusion network input. We discovered that decoding with confusion networks and unfiltered phrase-tables was not feasible. It was using enormous amounts of memory and time to cache and then decode all the possible translations. 1-best translations do not suffer nearly as much from this as having only one path through a sentence, drastically reduces the total number of possible input phrases. We discovered that we could speed up decoding enormously if we filtered the phrase table for only the top 100 translations for each input phrase. Most longer phrases have a reasonable number of translations, but some common phrases have enormous numbers of possible translations which are very poor. For instance, the source phrase "a" in the en-fr system, has 402 thousand translations. We therefore pruned the phrase table to eliminate the vast majority of these unhelpful translations, leaving us with only the top n most likely translations. We can see that translating with pruned phrase tables improves upon translating with just the lattice 1-best by 1.6 BLEU points. We can also see that changing the pruning limit does not affect the score very much, until a drastic limit of 1 is reached, where performance drops by 3.3 BLEU points. We further experimented by using the posterior probabilities on the lattice to prune the number of alternative arcs. We found that posterior pruning had a slightly negative effect, reducing the performance from confusion network input where we only pruned phrase tables, of between 0.2 and 0.3 BLEU points.

	BLEU
Absolute 1-best	17.0
Absolute 1-best Punctuated	16.1 (-0.9)
CN prune p.t. 100	11.1 (-5.9)

Table 5: Cased BLEU scores and decoding times for de-en models when tuned and tested on ASR output.

	en-fr	de-en
Edinburgh ASR system	22.45	14.92
IWSLT ASR system	23.00 (+0.55)	14.99 (+0.07)

Table 6: Official test 2013 cased BLEU results for 1-best SLT input. The Edinburgh ASR system input was our primary system.

The results of our de-en experiments are presented in Table 5. Here we see that the punctuated input does slightly worse, but because these are development data results, we do not rely upon them. We also see that confusion network results are much worse than the absolute 1-best.

1.5. Official Results

The results in Table 6 show the official results on our primary and contrastive submissions. The primary submissions used the absolute 1-best, unpunctuated ASR output of the Edinburgh system submissions. The contrastive submissions used the official IWSLT ASR output as input to the SMT decoder. The contrastive submissions did slightly better.

2. Machine Translation Systems

Our machine translation systems are based on our setup [1] that has been proven successful at the recent evaluation campaign at the Workshop on Statistical Machine Translation [20].

2.1. Baseline

The system uses the baseline Moses [8] phrase-based model [21] (as given in the example files for the experimental management system), with the following additions:

- limitation of phrase length to 5
- sparse domain indicator, lexical, phrase length, and count bin features [22]
- factored models for German–English and English–German
- source-side German compound splitting [23]
- cube pruning with pop limit 1000 for tuning, 5000 for testing [24]
- operation sequence model (OSM) with 4 additional supportive features: 2 gap based penalties, 1 distance based feature and 1 deletion penalty [25]
- batch k-best MIRA tuning [26]
- interpolated 5-gram KenLM language models [27]
- minimum Bayes risk decoding [28]
- no-reordering-over-punctuation heuristic [29]

In the IWSLT systems, we also used:

- compact phrase tables [30]
- filter out phrase translations with conditional probability of less than 0.0001
- hierarchical lexicalized reordering (mslr) [31]
- MADA tokenizer for source-side Arabic [32]
- Stanford Chinese segmenter [33]

We also tried hierarchical phrase-based models for Chinese, but did not achieve better results.

In addition to the data provided directly from the IWSLT organizers, we also included whenever applicable:

- Common Crawl parallel corpus, as provided by WMT 2013 [34]
- Europarl version 7 parallel corpus[1] [35]
- news commentary parallel corpus, as provided by WMT 2013
- news language model data provided by WMT 2013
- LDC Gigaword for French, Spanish, and English as output language

[1] http://www.statmt.org/europarl/

Language	Into English	From English
Arabic	24.8	7.6
Chinese	11.8	9.8
Dutch	32.8	26.5
Farsi	14.5	8.0
French	33.3	33.2
German	30.5	22.9
Italian	29.7	23.7
Polish	17.7	9.7
Portuguese	36.0	30.8
Romanian	31.7	21.1
Russian	19.1	13.1
Slovenian	24.7	18.0
Spanish	39.5	33.9
Turkish	13.5	7.2

Table 7: Baseline system performance for machine translation systems (Section 2.1): Cased BLEU scores on test2010 using NIST's mteval-v13a. Test on tune for Slovenian. Moses multi-bleu.perl for Chinese target.

We built systems for all language pairs of the IWSLT evaluation campaign. The quality scores (BLEU) of the resulting systems as measured on the development test set is given in Table 7.

2.2. Brown Cluster Language Models

As suggested by [36], we explored the use of Brown clusters [37]. We computed the clusters with GIZA++'s `mkcls` [38] on the target side of the parallel training corpus. Brown clusters are word classes that are optimized to reduce n-gram perplexity.

By generating the Brown cluster identifier for each output word, we are able to add an n-gram model over these identifiers as an additional scoring function. The inclusion of such an additional factor is trivial given the factored model implementation [39] of Moses. The n-gram model is trained on the target side of the TED corpus made available by the IWSLT organizers.

The motivation for using Brown clusters stems from the success of using n-gram models over part-of-speech and morphological tags and the lack of the required taggers and analyzers for many language pairs. Brown clustering induces word classes that are similar to part-of-speech tags (for instance, placing adjectives with the same inflection into one class), with some additional semantic grouping (for instance, grouping all color adjectives).

Results are shown in Table 8. While the Brown cluster sequence models do not help for some of the language pairs for which we have plentiful training data (French, Span-

Language	B_0	50	200	600	1000
Dutch	26.5	**26.7**	26.2	26.3	26.5
		+0.2	–0.4	–0.2	±0.0
French	33.2	33.3	**33.4**	33.1	33.1
		+0.1	+0.2	–0.1	–0.1
Polish	9.7	9.9	10.1	10.1	**10.4**
		+0.2	+0.4	+0.4	+0.7
Portuguese	30.8	31.6	32.2	32.4	**32.**4
		+0.8	+1.4	+1.6	+1.6
Russian	13.1	13.3	13.5	13.5	**14.0**
		+0.2	+0.4	+0.4	+0.9
Slovenian	18.0	**18.7**	18.6	17.7	18.0
		+0.7	+0.6	–0.3	±0.0
Spanish	34.1	34.3	**34.6**	34.5	34.0
		+0.2	+0.5	+0.4	–0.1
Turkish	7.2	7.4	7.5	**7.5**	7.5
		+0.2	+0.3	+0.3	+0.3

Table 8: Target sequence model ("language model") over Brown clusters: BLEU scores for different number of classes (50, 200, etc.) and improvement over the baseline (B_0). Translation from English only.

Language	B_0	50	200	600	1000
Dutch	26.5	**26.9**	26.5	26.6	26.5
		+0.2	+0.3	+0.3	±0.0
French	33.2	**33.8**	33.7	33.6	**33.8**
		+0.5	+0.3	+0.5	+0.7
Polish	9.7	10.1	**10.2**	**10.2**	10.1
		+0.2	+0.1	+0.1	–0.3
Portuguese	30.8	31.8	32.4	**32.3**	31.9
		–.02	+0.2	–0.1	–0.5
Russian	13.1	13.6	13.7	**13.8**	13.6
		+0.3	+0.2	+0.3	–0.4
Slovenian	18.0	18.6	**18.9**	18.2	18.0
		–0.1	+0.3	+0.5	±0.0
Spanish	34.1	**34.7**	34.6	34.6	34.6
		+0.4	±0.0	–0.1	+0.6
Turkish	7.2	7.3	7.3	**7.5**	**7.5**
		–0.2	–0.2	±0.0	±0.0

Table 9: Operation sequence model over Brown clusters: BLEU scores for different number of classes and improvement over the baseline of just using the Brown cluster sequence model ("language model"), as reported in Table 8.

ish, Dutch), we see good gains for others, especially for Portuguese and the morphologically rich Russian. For the first mentioned set of language models, we are also able to use part-of-speech tag sequence models (See Baseline systems in Table 10), but also without significant gains. Improvements are generally fairly robust independent of the number of clusters used.

2.3. Operation Sequence Models over Generalized Representations

The integration of the OSM model into phrase-based decoding [40, 41] addresses the problem of phrasal independence assumption since the model considers context beyond phrasal boundaries. However, due to data sparsity the model often falls back to very small context sizes. We investigated the use of generalized representations (pos, morphological analysis and word clusters) in the OSM model. The expectation is that given the sparse training data for many of the language pairs, defining this model over the more general word classes would lead to a model that is able to consider wider context and learn richer lexical and reordering patterns.

2.3.1. Brown Clusters

Using Brown clusters on the source side, enables us to use the cluster identifiers also for the operation sequence model. We added an operation sequence model over source and target clusters to each of the configurations of language and number of clusters reported in Table 8. We show improvements over each of these settings in Table 9. We generally see improvements, although there is no clear pattern with regard to number of clusters. The biggest gains are for the use of 1000 clusters for French and Spanish — the languages where the sequence model alone did not give much improvement.

We also tried using OSM models over different numbers of clusters simultaneously for English-to-{French, Spanish and Dutch} pairs. Small gain was observed in the case of English-to-Spanish as the best system improved from 34.7 to 35.0. No further gains were observed in the case of other two pairs. For each system, our official submission is the system with the best performance on the development test set.

2.3.2. POS and Morph Tags

We also tried using the OSM models over POS tags for English-to-{German, French, Spanish and Dutch} pairs. For German-English pairs we additionally used morphological tags on the German-side. We used LoPar [42] to obtain morphological analysis and POS annotation of German and MXPOST [43], a maximum entropy model for English POS tags. For other languages we used TreeTagger [44].

Model	English-German	German-English
Baseline	22.9	30.5
+OSM$_{(pos,pos)}$	23.2 +0.3	31.0 +0.5
+OSM$_{(pos,morph)}$	23.9 +1.0	31.2 +0.7
+OSM$_{all}$	24.2 +1.3	31.1 +0.6
	English-French	**English-Spanish**
Baseline	33.1	33.9
+OSM$_{(pos,pos)}$	33.0 -0.1	34.4 +0.5
	English-Dutch	
Baseline	26.6	
+OSM$_{(pos,pos)}$	26.6 ±0.0	

Table 10: Evaluating POS- and Morph-based OSM Models

The baseline systems shown in Table 10 used POS tags as an additional factor on source and target side and POS

target sequence model as an additional language model feature. English-to-German baseline used morphological target sequence model instead of POS sequence model. German-to-English baseline used morphological tags as additional factor on the source-side and POS tags on target-side.

Table 10 shows the effect of adding OSM models over POS and morph tags on top of the factor-augmented baseline systems. Adding an OSM model over [pos,morph] (source:pos,target:morph) combination gave best results for English-to-German. Similarly adding an OSM model over [morph,pos] (source:morph, target:pos) gave best results for German-to-English. Adding both the models simultaneously ($+\mathbf{OSM_{all}}$) gave further improvements for English-to-German but none for German-to-English pair.

Augmenting baseline systems with POS factors did not yield any improvement for English-to-{French, Spanish and Dutch} pairs. Adding POS-based OSM model did not help either, except for English-to-Spanish pair. Using cluster-ids instead of POS tags was found to be more useful for these pairs.

In a post-evaluation analysis we confirmed whether using generalized OSM models actually consider a wider contextual window than its lexically driven variant. We found that the probability of an operation is conditioned on less than a trigram in the OSM model over surface forms. In comparison OSM models over POS, morph or cluster-ids consider a window of roughly 4 previous operations thus considering more contextual information.

3. Summary

We have described our SLT and MT submissions to IWSLT-13 evaluation campaign. For SLT we experimented with different punctuation strategies and with using confusion network input. Punctuating the input as a separate preprocessing step is helpful, and improves en-fr results by 1.2 BLEU points. Working with confusion networks requires pruning of the phrase table so that the search space does not explode with very unlikely translations. We found that switching from the absolute 1-best ASR output to the hybrid lattice output from the ASR system had a very negative impact on translation (-5 BLEU points), which was surprising as the WER of the hybrid lattice system was not much worse. This suggests that WER is crucial for spoken language translation quality. Translating confusion networks however, improved translation quality by 1.2 BLEU points. Our MT submissions are based on the phrase-based pipeline as used in the recent WMT campaign. We additionally explored using Brown clusters, and linguistic annotations in factored-based phrase-translation model and the operation sequence model. Adding OSM model over POS and Morph tags gave improvements of +1.3 in English-to-German and +0.7 in German-to-English pairs. We showed the efficacy of using Brown clusters as additional factor in Phrase-based and OSM models. Our integration consistently improved the baseline system giving significant improvements in most cases. We obtained an average BLEU point improvements of up to +0.7 ranging from +0.3 to +1.6 translating from English to 8 European language pairs that contained a mixture of data sparse and morphologically rich languages. We also showed that using Brown clusters outperform POS tag in some language pairs. Table 11 show BLEU scores for our official submissions.

4. Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 287658 (EU BRIDGE) and grant agreement 288487(MosesCore).

5. References

[1] N. Durrani, B. Haddow, K. Heafield, and P. Koehn, "Edinburgh's machine translation systems for European language pairs," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 114–121. [Online]. Available: http://www.aclweb.org/anthology/W13-2212

[2] P. Bell, F. McInnes, S. Gangireddy, M. Sinclair, A. Birch, and S. Renals, "The UEDIN english ASR system for the IWSLT 2013 evaluation," in *Proc. IWSLT*, Heidelberg, Germany, 2013, submitted.

[3] J. Driesen, P. Bell, and S. Renals, "Description of the UEDIN System for German ASR," in *Proc. IWSLT*, Heidelberg, Germany, 2013, submitted.

[4] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2006, pp. 158–165.

[5] N. Bertoldi, R. Zens, and M. Federico, "Speech translation by confusion network decoding," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4. IEEE, 2007, pp. IV–1297.

[6] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: lattice-based word error minimization." in *Eurospeech*. Citeseer, 1999.

[7] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Proc. ASRU*, Big Island, Hawaii, US, December 2011.

[8] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. J. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for

	Into English			From English		
Language	$\mathbf{test_{11}}$	$\mathbf{test_{12}}$	$\mathbf{test_{13}}$	$\mathbf{test_{11}}$	$\mathbf{test_{12}}$	$\mathbf{test_{13}}$
Arabic	25.6	27.7	26.3	11.9	12.4	11.5
Chinese	16.1	14.2	15.3	19.8	18.1	18.6
Dutch	36.0	33.0	32.7	30.3	26.7	25.5
Farsi	19.2	15.9	15.1	12.3	10.2	9.5
French	–	–	–	40.6	41.2	38.5
German	–	–	25.5	27.1	22.5	24.0
Italian	30.2	29.6	34.9	24.4	25.3	29.2
Polish	21.7	18.5	20.9	13.1	10.5	11.5
Portuguese	39.0	40.6	37.3	33.6	34.9	33.2
Romanian	36.1	31.8	29.8	23.2	19.2	17.6
Russian	22.1	20.7	22.7	15.9	13.5	16.1
Slovenian	–	21.2	24.1	–	12.4	13.7
Spanish	37.1	30.8	39.1	33.2	26.8	34.7
Turkish	15.0	15.0	14.9	7.4	7.4	6.8

Table 11: Official Submissions (MT-Track) – Cased BLEU scores on test [2011-2013], using NIST's mteval-v13a

statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 177–180. [Online]. Available: http://www.aclweb.org/anthology/P/P07/P07-2045

[9] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[10] P. Koehn, "Europarl: A multilingual corpus for evaluation of machine translation," Unpublished, `http://www.isi.edu/~koehn/europarl/`, 2002.

[11] C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2012 workshop on statistical machine translation," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 10–51. [Online]. Available: http://cs.jhu.edu/~ccb/publications/findings-of-the-wmt12-shared-tasks.pdf

[12] J. Smith, H. Saint-Amand, M. Plamada, P. Koehn, C. Callison-Burch, and A. Lopez, "Dirt cheap web-scale parallel text from the Common Crawl," in *Proceedings of the 2013 Conference of the Association for Computational Linguistics (ACL 2013)*. Sofia, Bulgaria: Association for Computational Linguistics, July 2013. [Online]. Available: http://cs.jhu.edu/~ccb/publications/bitexts-from-common-crawl.pdf

[13] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 355–362.

[14] E. Hasler, P. Bell, A. Ghoshal, B. Haddow, P. Koehn, F. McInnes, S. Renals, and P. Swietojanski, "The UEDIN system for the IWSLT 2012 evaluation," in *Proc. International Workshop on Spoken Language Translation*, 2012.

[15] E. Hasler, B. Haddow, and P. Koehn, "Sparse lexicalised features and topic adaptation for smt," in *Proceedings of the International Workshop on Spoken Language Translation, Hong Kong, HK*, 2012.

[16] A. Stolke, "SRILM - an extensible language modeling toolkit," in *Proceedings of the International Conference on Spoken Language Processing*, 2002.

[17] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 187–193.

[18] M. Collins, P. Koehn, and I. Kučerová, "Clause restructuring for statistical machine translation," in *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2005, pp. 531–540.

[19] F. Weng, A. Stolcke, and A. Sankar, "Efficient lattice representation and generation," in *In Proc. of ICSLP*. Citeseer, 1998.

[20] O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2013 Workshop on Statistical Machine Translation," in *Proceedings of the*

Eighth Workshop on Statistical Machine Translation. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 1–44. [Online]. Available: http://www.aclweb.org/anthology/W13-2201

[21] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase based translation," in *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2003. [Online]. Available: http://acl.ldc.upenn.edu/N/N03/N03-1017.pdf

[22] D. Chiang, K. Knight, and W. Wang, "11,001 new features for statistical machine translation," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 218–226. [Online]. Available: http://www.aclweb.org/anthology/N/N09/N09-1025

[23] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proceedings of Meeting of the European Chapter of the Association of Computational Linguistics (EACL)*, 2003. [Online]. Available: http://acl.ldc.upenn.edu/E/E03/E03-1076.pdf

[24] L. Huang and D. Chiang, "Forest rescoring: Faster decoding with integrated language models," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 144–151. [Online]. Available: http://www.aclweb.org/anthology/P/P07/P07-1019

[25] N. Durrani, H. Schmid, and A. Fraser, "A joint sequence translation model with integrated reordering," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, June 2011, pp. 1045–1054. [Online]. Available: http://www.aclweb.org/anthology/P11-1105

[26] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 427–436. [Online]. Available: http://www.aclweb.org/anthology/N12-1047

[27] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, July 2011, pp. 187–197. [Online]. Available: http://www.aclweb.org/anthology/W11-2123

[28] S. Kumar and W. Byrne, "Minimum Bayes-risk decoding for statistical machine translation," in *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2004.

[29] P. Koehn and B. Haddow, "Edinburgh's submission to all tracks of the WMT2009 shared task with reordering and speed improvements to Moses," in *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Athens, Greece: Association for Computational Linguistics, March 2009, pp. 160–164. [Online]. Available: http://www.aclweb.org/anthology/W/W09/W09-0429

[30] M. Junczys-Dowmunt, "Phrasal rank-encoding: Exploiting phrase redundancy and translational relations for phrase table compression," *The Prague Bulletin of Mathematical Linguistics*, vol. 98, pp. 63–74, 2012.

[31] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, October 2008, pp. 848–856. [Online]. Available: http://www.aclweb.org/anthology/D08-1089

[32] N. Habash and F. Sadat, "Arabic preprocessing schemes for statistical machine translation," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. New York City, USA: Association for Computational Linguistics, June 2006, pp. 49–52. [Online]. Available: http://www.aclweb.org/anthology/N/N06/N06-2013

[33] P.-C. Chang, M. Galley, and C. D. Manning, "Optimizing Chinese word segmentation for machine translation performance," in *Proceedings of the Third Workshop on Statistical Machine Translation*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 224–232. [Online]. Available: http://www.aclweb.org/anthology/W/W08/W08-0336

[34] J. R. Smith, H. Saint-Amand, M. Plamada, P. Koehn, C. Callison-Burch, and A. Lopez, "Dirt cheap web-scale parallel text from the common crawl," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 1374–1383. [Online]. Available: http://www.aclweb.org/anthology/P13-1135

[35] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, Phuket, Thailand, September 2005.

[36] W. Ammar, V. Chahuneau, M. Denkowski, G. Hanneman, W. Ling, A. Matthews, K. Murray, N. Segall, A. Lavie, and C. Dyer, "The CMU machine translation systems at WMT 2013: Syntax, synthetic translation options, and pseudo-references," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 70–77. [Online]. Available: http://www.aclweb.org/anthology/W13-2205

[37] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computional Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[38] F. J. Och, "An efficient method for determining bilingual word classes," in *Ninth Conference the European Chapter of the Association for Computational Linguistics (EACL)*, June 1999, pp. 71–76.

[39] P. Koehn and H. Hoang, "Factored translation models," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 868–876. [Online]. Available: http://www.aclweb.org/anthology/D/D07/D07-1091

[40] N. Durrani, A. Fraser, and H. Schmid, "Model with minimal translation units, but decode with phrases," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 1–11. [Online]. Available: http://www.aclweb.org/anthology/N13-1001

[41] N. Durrani, A. Fraser, H. Schmid, H. Hoang, and P. Koehn, "Can markov models over minimal translation units help phrase-based smt?" in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 399–405. [Online]. Available: http://www.aclweb.org/anthology/P13-2071

[42] H. Schmid, "Lopar: Design and implementation," Institute for Computational Linguistics, University of Stuttgart, Bericht des Sonderforschungsbereiches "Sprachtheoretische Grundlagen fr die Computerlinguistik" 149, 2000.

[43] A. Ratnaparkhi, "Maximum entropy models for natural language ambiguity resolution," Ph.D. dissertation, University of Pennsylvania, Philadelphia, PA, 1998.

[44] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *New Methods in Language Processing*, ser. Studies in Computational Linguistics, D. Jones and H. Somers, Eds. London, GB: UCL Press, 1997, pp. 154–164. [Online]. Available: http://www.ims.uni-stuttgart.de/projekte/gramotron/PAPERS/MISC/NEMLAP97-TreeTagger.ps.gz

MSR-FBK IWSLT 2013 SLT System Description

Anthony Aue[1]*, Qin Gao*[1]*, Hany Hassan*[1]*, Xiaodong He*[1]*, Gang Li*[1]*, Nicholas Ruiz*[2]*, Frank Seide*[1]

[1]Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
anthaue@microsoft.com

[2]Fondazione Bruno Kessler
University of Trento
Trento, TN, Italy
nicruiz@fbk.eu

Abstract

This paper describes the systems used for the MSR+FBK submission for the SLT track of IWSLT 2013. Starting from a baseline system we made a series of iterative and additive improvements, including a novel method for processing bilingual data used to train MT systems for use on ASR output. Our primary submission is a system combination of five individual systems, combining the output of multiple ASR engines with multiple MT techniques. There are two contrastive submissions to help place the combined system in context. We describe the systems used and present results on the test sets.

1. Introduction

Our work for IWSLT 2013 [1] began with a baseline system that consisted of piping the 1-best output from FBK ASR system [2] through a phrase-based machine translation system [3]. We made a series of additive improvements to both the ASR and MT components, culminating in a combined system that significantly outperformed our baseline on the tst2010 test set. The biggest MT improvements came from augmenting the training data with data normalized to make it more similar to ASR output. The biggest ASR improvements came from using DNNs and doing speaker and language model adaptation.

We used three different ASR systems, which we will refer to in this paper as FBK, MSRA and MSRA-2. The FBK system is described in section 2.1. The MSRA and MSRA-2 systems are described in section 2.2.

We used four different MT systems, referred to hereafter as TREELET, PHRASE-BASED, PHONEME and OOD-PHONEME. The TREELET system is a tree-to-string translation system as described in [4]. The PHRASE-BASED system is a phrase-based machine translation system as described in [3]. The PHONEME system is a phrase-based system where the source side of the in-domain training data has been altered using a novel technique that makes it look more like ASR output. The technique used to alter the training data is novel. The OOD-PHONEME system is the same as the PHONEME system, but with the addition of out-of-domain normalized data.

Our primary submission was a system combination of five systems: FBK-TREELET, FBK-PHRASE-BASED, FBK-PHONEME, MSRA-PHONEME, and MSRA2-OOD-PHONEME. The system combination was performed using techniques described in [5].

In section 2 we discuss the ASR systems we used. Section 3 describes the work we did to insert punctuation into the ASR output. In section 4 we describe the machine translation systems we used. Results are discussed in section 5.

2. ASR Systems

Our system combination used the output from two different ASR engines. The first is the FBK engine described in [6]. The second is a system developed at Microsoft Research.

2.1. FBK ASR System

The FBK English speech recognizer is an HMM-based triphone large-vocabulary continuous-speech recognition system with acoustic models trained on both TED talks and out-of-domain data, such as the HUB4 broadcast news speech corpus. Lightly-supervised training is used to select reliable data from the TED talks, since the transcripts are inexact. The language model is constructed by filtering out all but 100 million words of the Gigaword and WMT 2013 out-of-domain corpora, as well as 2.7 million words from the provided in-domain data. Each corpus is used to train a distinct 4-gram language model, which are used to rescore the word graphs produced in the second recognition pass. Additionally, a linearly interpolation of the LMs is used for word graph rescoring. Word graph rescoring is used in the second recognition pass. System combination is performed with ROVER on the alternative rescoring methods. System performance on several IWSLT development and test sets are reported in Table 1. More details of the system can be found in [2].

2.2. MSR ASR System

The MSRA recognizer is an HMM-based triphone/trigram large-vocabulary continuous-speech recognition system that is fairly standard except that it uses a deep neural network for acoustic modeling—specifically a CD-DNN-HMM, or

context-dependent deep-neural network hidden Markov model [7, 8]. The system was developed out of a speaker-independent Switchboard system trained on 2000h of data (the SWBD and Fisher corpora), as described in [9]. That same model was used (with minor vocabulary tweaks) for a live demonstration of speech-to-speech [10], where one can get a subjective impression for its accuracy. In the following, we will describe how this system was adapted to the IWSLT task.

2.2.1. IWSLT Acoustic Model

The SWBD acoustic model is suboptimal for TED talks in that they are wideband recordings with a large variation of non-native accents. We switched training data to the TED-Lium collection [11], which consists of about 56000 utterances from 774 talks, which amounts to 118 hours of usable training speech after segmentation. The resulting DNN has 7 hidden layers of dimension 2048, and 9304 output classes.

The feature extraction was updated for wideband recordings and to reflect the latest experience w.r.t. DNNs. We used a raw 40-channel Mel-filterbank instead of PLPs, 10-th root non-linearity, and a wider frame window of 23 frames or about 1/4 of a second), instead of derivatives. This was followed by the usual mean-variance normalization.

The model training consisted of a first training round using the cross-entropy (CE) objective with regard to the "ground-truth" state-level time alignments created from a GMM starting model; realigning those using that DNN followed by further CE iterations; and then finally sequence training using the frame-smoothed maximum mutual information (FS-MMI) criterion [12].

The training process and model parameterization were chosen based on prior experience with different tasks without additional specific tuning for the IWSLT task.

2.2.2. IWSLT Language Model

The trigram language model was replaced by one trained on the provided "ASR LM Training Data English" since the SWBD language model was not admissible for this task, and interpolated with a second trigram language model trained on a large out-of-domain (OOD) collection (Gigaword, NewsCrawl, Europarl). Due to the vast size of this OOD collection, we aggressively pruned the OOD trigram to keep it at manageable size. The vocabulary was selected using a minimum word frequency of 40. The resulting vocabulary size was 110,813.

Table 1: Word error rates of FBK's primary English ASR submission on various IWSLT test sets.

	WER[%]				
System	**dev**$_{2010}$	**tst**$_{2010}$	**tst**$_{2011}$	**tst**$_{2012}$	**tst**$_{2013}$
Primary	17.0	15.7	13.6	16.2	23.2

2.2.3. Speaker Adaptation

Lastly, we used the fDLR feature transform for unsupervised speaker adaptation on each talk. fDLR, or feature-space discriminative linear regression [9], is a direct adaptation of the well-known fMLLR transform (also known as CMLLR), but using the discriminative cross-entropy criterion with back-propagation instead of maximum likelihood.

The fDLR process consists of a first-pass recognition that was configured to emit state-level alignments; inserting a virgin linear layer (the fDLR transform) at the bottom of the DNN stack; and then applying back-propagation to update the 40^2 tied fDLR parameters until convergence, using the first-pass recognition output as the "ground truth."

2.2.4. Results

Table 2 shows word-error rates (WERs) for three previous IWSLT test sets (dev2010, dev2012.en-sl, tst2010.en-fr). We see that the unmodified SWBD system performs 7 to 9 percentage points worse than the IWSLT-adapted system. We also see once again the benefit of the deep neural network: The WER of the TEDLium GMM starting model gets improved by the comparable DNN by a relative 30 to 37% (row "+ realign + CE training").

On top of that, the gain from sequence training is in the range of 3 to 6% relative. The row marked "sequence training" is the system labelled MSRA in the rest of this paper. The OOD LM gives us another 5 to 9%. Finally, fDLR speaker adaptation yields an up to 8% relative reduction. This is the system we will henceforth call MSRA-2. Despite doing no IWSLT-specific tuning (beyond swapping the training data), the resulting error rates are competitive with the best systems of IWSLT 2012.

Table 2: Word error rates of the MSRA recognizer on three previous IWSLT test sets for various configurations. The two rows in boldface are the MSRA and MSRA-2 systems, respectively.

	WER[%]		
System	**dev**$_{2010}$	**tst**$_{2010}$	**dev**$_{2012}$
SWBD DNN baseline	20.5	19.2	25.7
TEDLium, GMM start	25.0	25.5	29.4
+ DNN, CE-trained	17.6	15.7	18.7
+ realign + CE training	17.4	15.6	18.6
+ sequence training	**16.3**	**15.1**	**17.8**
+ OOD LM	15.2	13.8	16.8
+ speaker adaptation	**14.6**	**12.9**	**15.5**

3. Punctuation Insertion

3.1. Punctuation restoration strategies

Punctuation restoration is an important task for Spoken Language Translation (SLT). Speech recognition systems provide neither punctuation nor sentence boundaries in the pro-

duced text. In this work, the sentence boundaries are provided by the IWSLT evaluation task; therefore we focus only on intra-sentence punctuation restoration.

Generally, there are three strategies for punctuation restoration for SLT.

1. Inserting punctuation on the output of the ASR system before feeding it as the input to the machine translation system. In this case, we can use conventional machine translation systems trained on punctuated text in both source and target languages.

2. Handling punctuation insertion as part of the translation process, where translation is done from ASR-like unpunctuated text as the source and fully punctuated text as the target.

3. Proceeding as in the second strategy but producint unpunctuated target text and trying to restore punctuation on the produced target text.

Previous work in [13] showed that the first strategy provides the best results with machine translation quality. Therefore, in the current work we choose the first strategy where we process the ASR output to restore intra-sentence punctuation as a preprocessing step before translation.

3.2. The Approach

Using SMT for punctuation restoration was introduced in [14], where a phrase-based translation system was trained to translate from unpunctuated source text to punctuated target text with pseudo bilingual data obtained by removing punctuation from the source side and leaving the target side punctuated. They showed significant improvement on the IWSLT-2007 evaluation when they deployed this approach as a post-processing step for restoring punctuation for unpunctuated target text. More recently, [13] evaluated the same approach as a preprocessing step for ASR output and as a post-processing step for unpunctuated target translation. They found that using it as a preprocessing step is significantly better than post-processing. In this work, we adopt the same approach as a preprocessing step.

Our system is a phrase-based MT system; we use a monotonic decoder with no reordering and no distortion penalty. The language model is a 5-gram LM trained on the target side of the parallel data.

3.3. Data and data preparation

Our training data is English data from IWSLT out-of-domain data. We selected 26M sentences of the English side of the data from Europarl and News Broadcast. We processed the data to remove all punctuation except for periods, commas, semi-colons, question marks, apostrophes and exclamation points. This processed data represents the target side of our MT system. The source side of the translation data is obtained by removing the sentence boundary punctuation (periods, commas, semi-colons, question marks and exclamation points). Therefore, the purpose of the system is to produce punctuated text form unpunctuated text within the sentence. We use two sets of 5000 sentences from the TED talks data as our development and test sets for the punctuation restoration system.

BLEU	**Case Insensitive**	**Case Sensitive**
Baseline	22.5	20.83
Punctuation Restored	24.42 (+1.92)	22.71 (+1.88)

Table 3: Punctuation Restoration Results

3.4. Results

We evaluate the system on the translation task directly; where we restore punctuations and compare the effect of restoring the punctuation on the overall translation quality. We use the English-French translation task; where the baseline is translating without punctuation restored. The table shows the translation results with and without punctuation for the English-French translation task. The baseline has no punctuation restored. The system shows significant improvement of 8.5% over the baseline in terms of overall BLEU score.

4. MT Systems

This section describes the various machine translation systems we used.

4.1. Training Data

We used the same training data to train all of our machine translation systems. For in-domain parallel data, we used the TED corpus provided by the competition. Out-of-domain parallel data was

1. Gigaword
2. MultiUN
3. Europarl V7
4. Parallel News commentary V8
5. WMT 2013 News Commentary (Common Crawl)

Data to build the French target language model was

1. News Commentary V8
2. News Crawl
3. French Gigaword V3
4. European Language Newspaper Text LDC95T11

4.2. Baseline System

Our baseline system is a typical phrase-based statistical machine translation system. Details of the system are described in [3]. The decoder is very similar to the one used by Moses [15].

4.3. Treelet System

In addition to our phrase-based baseline system, we also used a syntax-based tree to string MT system, as described in [4]. Although the BLEU score of this system individually is somewhat lower than that of the baseline phrase-based system, it is able to capture certain phenomena that are hard to capture in phrase-based systems. It is thus a very useful component for system combinations.

4.4. Phoneme-motivated Text Normalization

Machine translation relies heavily on the data it uses in training. Simply training a MT system on text corpora and applying it to spoken language translation creates a search space that is inaccessible by the output of the ASR system. Therefore, it is very important to have a representative training corpus for translating spontaneous speech, instead of written text. Unfortunately, bilingual spontaneous speech corpora of sufficient size for high-quality MT are not widely available. We chose to adapt our written training data to look more like speech.

The ASR output deviates from written text in the following ways:

1. Delinquencies, such as restarts and word deletions.
2. Tokens in their pronounced form. For example, the token *1990* can have different pronounced forms based on its context; namely "nineteen ninety" or "one thousand nine hundred ninety". Other symbols may also be pronounced or ignored depending on the context.
3. ASR errors. These errors may come from homophone confusions, e.g. *theirs* vs. *there's*; reference words not appearing in the lexicon (OOV words), misrecognized phonemes, e.g. *is* instead of *its*; and biases from the language model. In the case of OOV errors, the words not appearing in the ASR lexicon are substituted with phonetically similar in-vocabulary words.

We consider the ASR system as a channel that maps transcripts into recognition results. Were there training data that maps speech recognition outputs to translations, we could train a machine translation system without relying on text corpora. Since this is rarely the case, we attempt to adapt the MT data into ASR-like output to anticipate both potential ASR errors and text normalizations that transform texts into a canonical form.

To motive our work, let's consider a concrete example of ASR output:

```
Transcript: And there are...
ASR output: And their are...
Reference : Et il y a...
MT output : Et leur font...
```

We can see that *there* and *their* are commonly confused homophones. While this error may occur frequently in ASR output, a machine translation system that is trained on written text in professional domains will not encounter this error and will not have sufficient statistics to translate *their are* as *il y a*. Therefore, in our work we try to simulate the recognition behavior of the ASR system by converting written text into the phoneme space, and then map back to the text space using a phrase-based MT system trained using components from the system we want to simulate.

4.4.1. System Configuration

Inspired by the expositions of [16, 17], we first normalize each word in FBK's ASR lexicon into a phoneme sequence by performing text-to-speech (TTS) analyses with an in-house synthesizer. The phoneme sequences and their target lexical forms are used respectively as source and target parallel training data for a monotonic phoneme-to-word phrase-based MT system. We use two 4-gram LMs from FBK's IWSLT 2012 primary submission [6], which were trained with modified Kneser-Ney smoothing [18] on TED and WMT data. Due to the small amount of training data, we assign uniform forward and backward phrase probabilities to each phoneme sequence to word mapping. We omit lexical probabilities.

With the aforementioned components, we can now tune a phrase-based machine translation system that translates from the *actual* phoneme sequence into *ASR text* output. The optimization can be done by randomly sampling a small development set from the 1-best ASR outputs on dev2010 from FBK's IWSLT 2012 primary submission. The corresponding transcripts are used as input in MERT. We apply the tuned phoneme-to-word translation system to all the training data and concatenate the synthesized bitexts with existing written bitexts as additional training data. Figure 1 provides a graphical depiction of the pipeline.

As we mentioned, the method we proposed tries to address the problem of ASR errors. The generated bitext has the following properties:

- All the numerals and symbols are converted into their pronounced form.
- Homophone errors and combination errors are injected into the new bitext.
- The text will not contain any OOVs that don't appear in the ASR system's lexicon. OOV words will be mapped to their most likely alternatives.

4.4.2. Expanding Pronunciation Hypotheses

Since our TTS analyzer in the .NET framework provides the single best phoneme sequence for an utterance, we expanded the phoneme sequences generated for each word in the ASR lexicon by performing TTS analysis on each transcript line in the TED training data and aligning the phoneme sequences to each corresponding word. Pronunciations for word entries not appearing in the ASR lexicon are ignored. We also

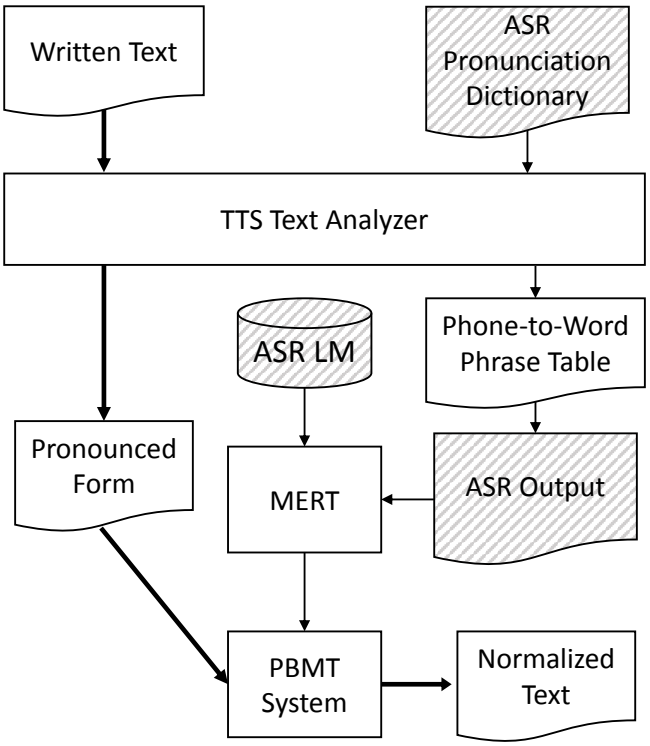


Figure 1: Phonetic normalization pipeline

captured count statistics on each pronunciation sequence to word mapping. These counts were used to rank the forward and backward probabilities of the pronunciation phrase table by $(1/1+r)$, where r is the rank of the pronunciation mapping.

4.4.3. Results

We compare the normalization techniques described above against a baseline MT system containing only un-normalized text. Our first system (TED norm) performs the normalization technique in 4.4.1, using uniform phrase translation probabilities. In the second system, we normalize all of the MT training data and use the phrase-based translation probability features estimated from the TED data, as described in 4.4.2.

Both the original and improved channel model results are provided in Table 4 using FBK's output.

4.5. System Combination

In testing, we combined outputs from the five single systems using the incremental indirect hidden Markov model (IHMM) proposed in [5, 19], which has been shown to give superior performance in several MT benchmark tests [20]. The parameters of the IHMM are estimated indirectly from a variety of sources including semantic word similarity, surface word similarity, and a distance-based distortion penalty. The pairwise IHMM was extended to operate incrementally in [19], where the confusion network is initialized by forming a simple graph with one word per link from the skeleton hypothesis, and each remaining hypothesis is aligned with the partial confusion network. This allows words from all previous hypotheses be considered as matches and leads to better performance compared to the pairwise IHMM. The incremental IHMM is also more computationally efficient than fully joint optimization methods such as [21], and provides a good trade-off between accuracy and runtime cost. In our implementation, each of these five systems produces a 10-best output for system combination. The semantic word similarity of the IHMM is derived from the French/English word translation probabilities learned on the TED parallel training data using the word-dependent HMM-based alignment method proposed in [22]. The language model is a trigram LM trained on the French side of the TED parallel data. The system combination parameters are tuned on the first half of the IWSLT tst2010 set, while the second half is reserved as the devtest set.

Normalization	**tst**$_{2010}$
None	23.60
TED-only	24.50
ALL	25.03

Table 4: Evaluation results for text normalization. RAW refers to un-normalized training corpora. Normalization techniques use TTS analysis to convert input data into phoneme sequences, followed by channel modeling trained from the ASR lexicon (LEX) and optionally the TED training data to generate normalized text.

5. Results

Here we present the results of testing our various systems on test sets.

5.0.1. Test Data

Because we observed mismatches between the dev2010 and tst2010 test sets which made dev2010 unsuitable for use in tuning our system combination, we decided to use half of tst2010 as a development test set and the other as a held-out test set. Throughout the rest of the paper we will refer to these sets as tst2010-dev and tst2010-test.[1] It should be noted that only the system combination parameters were trained on the tst2010-dev. None of the individual systems used tst2010-dev for training or parameter tuning, so results on these sets are valid test results. However we have chosen to report results for the individual systems on the two halves of tst2010 separately in order to make them comparable with the results of the combined system. As the reader will note, the results on the two halves are generally very close. Reported results are case-sensitive, punctuation-sensitive BLEU.

[1] Tst2010-dev contains talks 767, 769, 779, 783, and 785, while tst2010-test contains talks 790, 792, 799, 805, 824, and 827.

System	tst2010-dev	tst2010-test
fbk.baseline	22.05	21.57
fbk.phoneme	21.75	21.85
fbk.ood-phoneme	**22.16**	**22.52**
fbk.treelet	20.41	20.8
msra.baseline	22.04	21.83
msra.phoneme	22.18	22.09
msra.ood-phoneme	22.67	22.74
msra.treelet	20.92	21.19
msra-2.baseline	22.88	22.41
msra-2.phoneme	23.11	22.84
msra-2.ood-phoneme	**23.46**	**23.61**
msra-2.treelet	21.69	22.15
syscombo3 (First three)	22.9	22.41
syscombo5 (all five)	**24.4**	**24.08**

Table 5: BLEU Results from all systems on tst2010-dev and tst2010-test. Our primary submission was syscombo5. Contrastive1 msra-2.ood-phoneme, our best single system. Contrastive2 is fbk.ood-phoneme

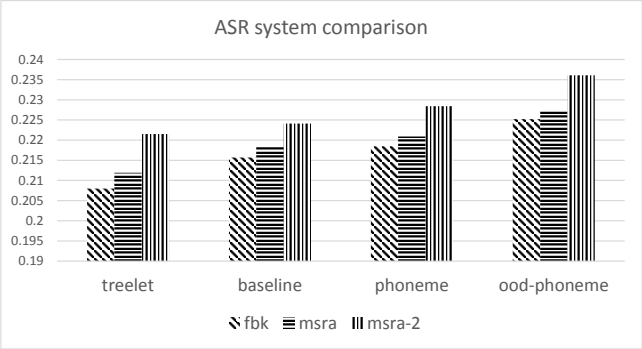


Figure 2: BLEU Results by ASR system

5.0.2. Test results

In Table 5, we report results for each ASR system + MT system combination. In figure 2, we can see the BLEU scores for each ASR system, grouped by translation system. In figure 3, we can see BLEU scores for each MT system, grouped by ASR system. The trends are very clear. On the ASR side, the benefits of using DNNs, speaker adaptation and a large out-of-domain LM are quite clear and robust across MT systems. For the MT systems, the advantage of adapting the training data with the phoneme method is also clear, with OOD-PHONEME systems outperforming systems with only in-domain adapted data across the board. System combination of 5 systems buys about 1 BLEU point on top of the best single system.

Table 6 contains our results on the official SLT test set (tst2013) as well as the progress test sets tst2010, tst2011 and tst2012. As the reader can see, our results on tst2010 and tst2012 were very different from those on tst2011 and tst2013. On tst2010, syscombo5 (our primary submission)

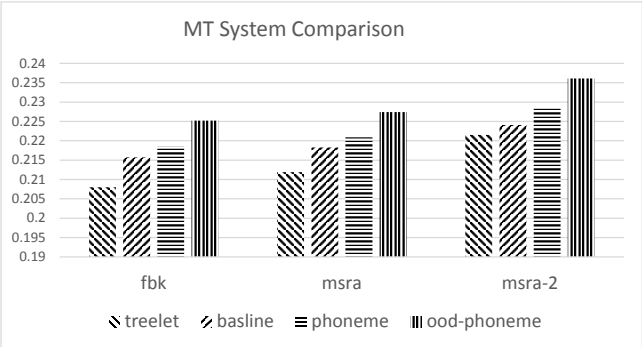


Figure 3: BLEU Results by MT system

scores a full BLEU point above msra-2.ood-phoneme, which is in turn almost a full point above fbk.ood-phoneme (contrastive2). Syscombo5 also scores highest on tst2012. Conversely, fbk.ood-phoneme scores higher than syscombo5 on tst2013 (by nearly two BLEU points!) and on tst2011. The odd-yeared and even-yeared test sets seem to show significant signals pointing in different directions. We have thus far been unable to find a good explanation for this discrepancy. There are several possible factors.

Regarding the ordering of two phoneme-normalized systems (fbk.ood-phoneme vs. msra-2.ood-phoneme), it is worth noting that the data normalizations for both systems were derived from the FBK dictionaries and language models. This suggests an obvious bias in favor of fbk.ood-phoneme over msra-2.ood-phoneme. Perhaps the effects of this bias were weaker in the tst2010 set than in the other test sets. We plan to train a normalizing system using the vocabulary from the msra-2 system in order to test the significance of this effect.

The difference in the ordering of the syscombo5 system in relation to the other systems is even starker and more difficult to explain. Strong distributional similarity between tst2010-dev and tst2010-test might have led to overfitting on that test set. However this seems unlikely given that the sets of talks contained in the two splits are disjoing. Furthermore, that hypothesis fails to explain the very strong performance of syscombo5 on tst2012.

	Metric	tst_{2010**}	tst_{2011}	tst_{2012}	tst_{2013}
P	BLEU	24.08	27.21	29.92	22.42
	TER	–	0.5622	0.5330	0.637
C_1	BLEU	23.61	26.72	–	20.96
	TER	–	0.5706	–	0.654
C_2	BLEU	22.16	27.55	29.47	24.36
	TER	–	0.5647	0.5358	0.599

Table 6: Results of submitted English-French runs evaluated on the IWSLT TED test sets. Note re. tst2010**: Because we used the first half of tst2010 as a development set for system combination in our primary submission, we report results only for the second half of tst2010. As one can see in Table 5, the BLEU scores for the two halves are generally very close, so this is a decent proxy for the whole test set.

6. References

[1] M. Cettolo, J. Niehues, S. Stker, L. Bentivogli, and M. Federico, "Report on the 10th IWSLT Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Heidelberg, Germany, 2013.

[2] D. Falavigna, R. Gretter, F. Brugnara, and R. H. Serizel, "FBK @ IWSLT 2013 - ASR tracks," in *Proceedings of the International Workshop on Spoken Language Translation*, Heidelberg, Germany, 2013.

[3] R. C. Moore and C. Quirk, "Faster Beam-search Decoding for Phrasal Statistical Machine Translation," in *In Proceedings of MT Summit XI*. Citeseer, 2007.

[4] A. Menezes and C. Quirk, "Syntactic Models for Structural Word Insertion and Deletion during Translation," in *EMNLP*. ACL, 2008, pp. 735–744.

[5] X. He, M. Yang, J. Gao, P. Nguyen, and R. Moore, "Indirect-HMM-based Hypothesis Alignment for Combining Outputs from Machine Translation Systems," in *EMNLP*. ACL, 2008, pp. 98–107.

[6] D. Falavigna, R. Gretter, F. Brugnara, and D. Giuliani, "FBK @ IWSLT 2012 - ASR track," in *Proceedings of the International Workshop on Spoken Language Translation*, Hong Kong, China, 2012.

[7] D. Yu, L. Deng, and G.Dahl, "Roles of Pretraining and Fine-Tuning in Context-Dependent DNN-HMMs for Real-World Speech Recognition," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[8] F. Seide, G. Li, and D. Yu, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks," in *Interspeech*. icml.cc / Omnipress, 2011.

[9] F. Seide, G. Li, X. Chen, and D. Yu, "Feature Engineering in Context-Dependent Deep Neural Networks for Conversational Speech Transcription," in *ASRU*, D. Nahamoo and M. Picheny, Eds. IEEE, 2011, pp. 24–29.

[10] R. Rashid, "Microsoft Research Shows a Promising New Breakthrough in Speech Translation Technology," http://blogs.technet.com/b/next/archive/2012/11/08/microsoft-research-shows-a-promising-new-breakthrough-in-speech-translation-technology.aspx, 2012, [Online; accessed 31-Oct-2013].

[11] A. Rousseau, P. Deléglise, and Y. Estève, "TED-LIUM: an Automatic Speech Recognition Dedicated Corpus," in *LREC*, N. Calzolari, K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds. European Language Resources Association (ELRA), 2012, pp. 125–129.

[12] H. Su, G. Li, D. Yu, and F. Seide, "Error Back Propagation for Sequence Training of Context-Dependent Deep Networks for Conversational Speech Transcription," in *ICASSP*, 2013.

[13] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling Punctuation Prediction as Machine Translation," in *International Workshop on Spoken Language Translation (IWSLT)*, 2011.

[14] H. Hassan, Y. Ma, and A. Way, "Matrex: the DCU Machine Translation System for IWSLT 2007," in *International Workshop on Spoken Language Translation (IWSLT)*, 2007.

[15] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *ACL*, J. A. Carroll, A. van den Bosch, and A. Zaenen, Eds. The Association for Computational Linguistics, 2007.

[16] Q. F. Tan, K. Audhkhasi, P. G. Georgiou, E. Ettelaie, and S. S. Narayanan, "Automatic Speech Recognition System Channel Modeling," in *INTERSPEECH*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds. ISCA, 2010, pp. 2442–2445.

[17] K. Sagae, M. Lehr, E. T. Prud'hommeaux, P. Xu, N. Glenn, D. Karakos, S. Khudanpur, B. Roark, M. Saraclar, I. Shafran, D. M. Bikel, C. Callison-Burch, Y. Cao, K. Hall, E. Hasler, P. Koehn, A. Lopez, M. Post, and D. Riley, "Hallucinated N-best Lists for Discriminative Language Modeling," in *ICASSP*. IEEE, 2012, pp. 5001–5004.

[18] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," in *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, 1996, pp. 228–235.

[19] C.-H. Li, X. He, Y. Liu, and N. Xi, "Incremental HMM Alignment for MT System Combination," in *ACL/IJCNLP*, K.-Y. Su, J. Su, and J. Wiebe, Eds. The Association for Computer Linguistics, 2009, pp. 949–957.

[20] A.-V. Rosti, X. He, D. Karakos, G. Leusch, Y. Cao, M. Freitag, S. Matsoukas, H. Ney, J. Smith, and B. Zhang, "Review of Hypothesis Alignment Algorithms for MT system Combination via Confusion Network Decoding," in *Proceedings of NAACL-HLT workshop on SMT (WMT)*, 2012.

[21] X. He and K. Toutanova, "Joint Optimization for Machine Translation System Combination," in *EMNLP*. ACL, 2009, pp. 1202–1211.

[22] X. He, "Using Word-Dependent Transition Models in HMM Based Word Alignment for Statistical Machine Translation," in *ACL-WMT*, 2007.

[23] *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 2008.

Improving machine translation into Chinese by tuning against Chinese MEANT

Chi-kiu Lo, Meriem Beloucif, Dekai Wu

HKUST
Human Language Technology Center
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
{jackielo|mbeloucif|dekai}@cs.ust.hk

Abstract

We present the first ever results showing that Chinese MT output is significantly improved by tuning a MT system against a semantic frame based objective function, MEANT, rather than an n-gram based objective function, BLEU, as measured across commonly used metrics and different test sets. Recent work showed that by preserving the meaning of the translations as captured by semantic frames in the training process, MT systems for translating into English on both formal and informal genres are constrained to produce more adequate translations by making more accurate choices on lexical output and reordering rules. In this paper we describe our experiments in IWSLT 2013 TED talk MT tasks on tuning MT systems against MEANT for translating into Chinese and English respectively. We show that the Chinese translation output benefits more from tuning a MT system against MEANT than the English translation output due to the ambiguous nature of word boundaries in Chinese. Our encouraging results show that using MEANT is a promising alternative to BLEU in both evaluating and tuning MT systems to drive the progress of MT research across different languages.

1. Introduction

We present the first ever results of tuning a MT system against a semantic frame based objective function in order to produce a more adequate Chinese translation output. We compare the performance of our systems in IWSLT 2013 TED talk MT tasks on Chinese-English and English-Chinese translation with that of the baseline SMT systems tuned against BLEU. We show that the improvement of tuning a MT system against MEANT on Chinese translation output is more significant because of the nature of ambiguous word boundaries in Chinese. Our encouraging results show that using MEANT is a promising alternative to BLEU in evaluating and tuning MT systems to drive the progress of MT research across different languages.

In the past decade, the progress of MT research is predominantly driven by the fast and cheap n-gram based MT evaluation metrics, such as BLEU [1], which assume that a good translation is one that shares the same lexical choices as the reference translation. Despite enforcing fluency, it has been established that these metrics do not enforce translation utility adequately and often fail to preserve meaning[2, 3]. Unlike BLEU, or other n-gram based MT evaluation metrics, the MEANT family of metrics [4, 5, 6] adopt at outset the principle that a good translation is one from which humans can successfully understand at least the central meaning of the input sentence as captured by the basic event structure— "*who did what to whom, when, where and why"*[7]. [6]MEANT measures similarity between the MT output and the reference translations by comparing the similarities between the semantic frame structures of output and reference translations. For evaluating English translations, we have shown that MEANT correlates better with human adequacy judgment than commonly used MT evaluation metrics, such as BLEU [1], NIST [8], METEOR [9], CDER [10], WER [11], and TER [12].

We recently showed that the translation adequacy across different genres (ranging from formal news to informal web forum) is improved by replacing surface oriented metrics like BLEU or TER with a semantic frame based objective function, MEANT, when tuning the parameters of MT systems [13, 14]. However, the question of whether the same approach of tuning MT systems against a semantic objective function might improve translation adequacy when translating into other

languages, such as Chinese, is left unanswered.

Although there exists no studies on correlation between human adequacy judgement and MEANT scores on Chinese output, we hypothesize that the benefits of tuning against MEANT that we see for English: better adequacy and fluency carries over into Chinese. It is because a high MEANT score is contingent on correct lexical choices as well as getting the syntactic and semantic structures right, which is language independent.

The proposed approach of incorporating semantic information into SMT by tuning the model against a semantic frame based evaluation metric is independent of assumptions about the underlying translation model architecture. Therefore, we show that MT systems from different SMT approaches, flat phrase-based and hierarchical phrase-based, both benefit from the semantic information incorporated through our approach.

2. Related work

2.1. MT evaluation metrics

N-gram or edit distance based metrics such as BLEU [1], NIST [8], METEOR [9], CDER [10], WER [11], and TER [12] do not correctly reflect the similarity of the basic event structure— "*who did what to whom, when, where and why*"— of the input sentence. In fact, a number of large scale meta-evaluations [2, 3] report cases where BLEU strongly disagrees with human judgments of translation adequacy.

This has caused a recent surge of work on developing MT evaluation metrics that would outperforms BLEU in correlation with human judgment. AMBER [15] shows a high correlation with human adequacy judgment [16], however, it is very hard to interpret and indicate what errors the MT systems are making.

ULC [17, 18] is an automatic metric that incorporates several semantic similarity features and shows improved correlation with human judgement of translation quality [19, 17, 20, 18] but no work has been done towards tuning an SMT system using a pure form of ULC perhaps due to its expensive run time. Similarly, SPEDE [21] is an integrated probabilistic FSM and probabilistic PDA model that predicts the edit sequence needed for the MT output to match the reference. Sagan [22] is a semantic textual similarity metric based on a complex textual entailment pipeline. These aggregated metrics require sophisticated feature extraction steps; contain several dozens of parameters to tune and employ expensive linguistic resources, like WordNet and paraphrase tables. Like ULC, these metrices are not useful in the MT system development cycle for tuning due to expensive running time. The metrics themselves are also expensive in training and tuning due to the large number of parameters that need to be estimated.

ROSE [23] is a weighted linear model of shallow linguistic features which is cheaper in run time but still contains several dozens of weights that need to be tuned, which makes it hard to port the metric to different domains. TINE [24] is an automatic recall-oriented evaluation metric which aims to preserve the basic event structure. However, it performs comparably to BLEU and worse than METEOR on correlation with human adequacy judgment.

In contrast, there is very little work on designing MT evaluation metrics for evaluating Chinese or other languages with ambiguous word boundaries. For instance, studies show that simply adapting the commonly used MT evaluation metrics to evaluate Chinese on character-level showed a higher correlation with human judgment than the original word-level evaluation metrics [25]. Later, TESLA-CELAB is introduced as a hybrid character-level and word-level MT evaluation metric for evaluating Chinese [26]. Although TESLA-CELAB correlates significantly better with human judgment for evaluating Chinese than BLEU, no work has been done towards tuning an SMT system for translating into Chinese using it.

2.2. The MEANT family of metrics

MEANT [6], which is the weighted f-score over the matched semantic role labels of the automatically aligned semantic frames and role fillers, outperforms BLEU, NIST, METEOR, WER, CDER and TER in correlating with human adequacy judgment. MEANT is easily portable to other languages requiring only an automatic semantic parser and a large monolingual corpus in the output language for identifying the semantic structures and the lexical similarity between the semantic role fillers of the reference and translation.

Precisely, MEANT is computed as follows:

1. Apply an automatic shallow semantic parser to both the references and MT output. (Figure 1 shows examples of automatic shallow semantic parses on both reference and MT output.)

2. Apply the maximum weighted bipartite matching algorithm to align the semantic frames between

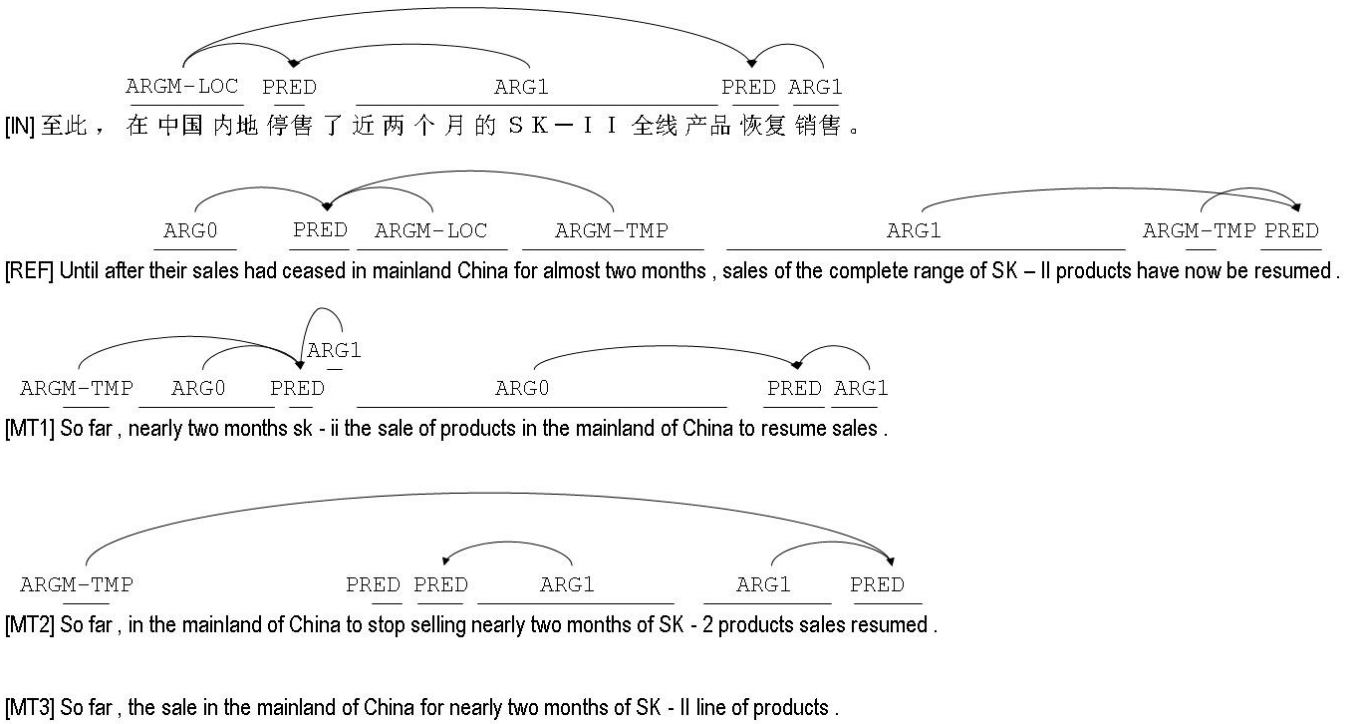


Figure 1: Examples of automatic shallow semantic parses. The input is parsed by a Chinese automatic shallow semantic parser. The reference and MT output are parsed by an English automatic shallow semantic parser. There are no semantic frames for MT3 since there is no predicate.

the references and MT output by the lexical similarities of the predicates.

3. For each pair of aligned semantic frames,
   (a) Determine the similarity of the semantic role fillers using Lexical similarity scores.
   (b) Apply the maximum weighted bipartite matching algorithm to align the semantic role fillers between the reference and MT output according to their lexical similarity.
4. Compute the weighted f-score over the matching role labels of these aligned predicates and role fillers acording to the mathematical definitions in the following.

$$
\begin{aligned}
M_{i,j} &\equiv \text{total \# ARG j of aligned frame i in MT} \\
R_{i,j} &\equiv \text{total \# ARG j of aligned frame i in REF} \\
S_{i,\text{pred}} &\equiv \text{similarity of predicate in aligned frame i} \\
S_{i,j} &\equiv \text{similarity of ARG j in aligned frame i} \\
w_{\text{pred}} &\equiv \text{weight of similarity of predicates} \\
w_j &\equiv \text{weight of similarity of ARG j}
\end{aligned}
$$

$$
\begin{aligned}
m_i &\equiv \frac{\text{\#tokens filled in aligned frame i of MT}}{\text{total \#tokens in MT}} \\
r_i &\equiv \frac{\text{\#tokens filled in aligned frame i of REF}}{\text{total \#tokens in REF}}
\end{aligned}
$$

$$
\text{precision} = \frac{\sum_i m_i \frac{w_{\text{pred}} S_{i,\text{pred}} + \sum_j w_j S_{i,j}}{w_{\text{pred}} + \sum_j w_j M_{i,j}}}{\sum_i m_i}
$$

$$
\text{recall} = \frac{\sum_i r_i \frac{w_{\text{pred}} S_{i,\text{pred}} + \sum_j w_j S_{i,j}}{w_{\text{pred}} + \sum_j w_j R_{i,j}}}{\sum_i r_i}
$$

where m_i and r_i are the weights for frame, i, in the MT/REF respectively. These weights estimate the degree of contribution of each frame to the overall meaning of the sentence. $M_{\text{i,j}}$ and $R_{\text{i,j}}$ are the total counts of argument of type j in frame i in the MT and REF respectively. $S_{i,\text{pred}}$ and $S_{i,j}$ are the lexical similarities (as computed based on a context vector model) of the predicates and role fillers of the arguments of type j between the reference translations and the MT output. The weights w_{pred} and w_j are the weights of the lexical similarities of the predicates and role fillers of the arguments of type j between the reference translations and the MT output. There is a total of 12 weights for the set

of semantic role labels in MEANT as defined in [27]. For MEANT, w_{pred} and w_j are determined using supervised estimation via a simple grid search to optimize the correlation with human adequacy judgments [4]. For UMEANT, w_{pred} and w_j are estimated in an unsupervised manner using relative frequency of each semantic role label in the reference translations. UMEANT can thus be used when when human judgments on adequacy of the development set are unavailable [5].

2.3. Tuning against better evaluation metrics

Previous works show that tuning MT system against better evaluation metrics improve the translation quality [28, 29]. Recent studies [13, 14] also shows that tuning MT system against MEANT produces more robustly adequate translations than the common practice of tuning against BLEU or TER across different data genres, such as formal newswire text, informal web forum text and informal public speech. Therefore, we believe that tuning MT systems against MEANT would improve the adequacy on Chinese MT output.

3. Experimental setup

In this section, we describe the details of our systems for the English-Chinese and Chinese-English TED talk MT tasks in terms of data, preprocesing, SMT pipeline and MEANT settings.

3.1. Data and preprocessing

Since our focus in this evaluation campaign is running contrastive experiments on tuning different MT systems against different MT evaluation metrics, we have deliberately constrained our training data to in-domain data only. For the translation model we have only used the officially released parallel training data, while for the language model we have only used the output side of the released training data. Similarly, no additional data was used as a part of development set other than the officially released development set. In order to test the consistency of the experimental results the test sets of IWSLT 2011 and 2012 were used in addition to the IWSLT 2013 test set. We perform minimal preprocessing on the training data running a maximum entropy Chinese segmenter [30] along with numex/timex segmenter on the Chinese data and punctuation tokenization and true casing on the English data.

3.2. SMT pipeline

With the goal of improving MT utility by using MEANT as an objective function to drive minimum error rate training (MERT) [31] of state-of-the-art MT systems, we setup our baseline using Moses [32], an off-the-shelf translation toolkit. In this paper we have two baselines: a flat phrase-based MT and a hierarchical phrase-based MT [33]. This allows us to use Moses to compare the performance of MEANT-tuned systems in these two different MT paradigms.

The language models are trained using the SRI language model toolkit [34]. For both translation tasks, we used a 6-gram language model. We use ZMERT [35] to tune the baseline because it is a widely used, highly competitive, robust, and reliable implementation of MERT that is also fully configurable and extensible with regard to incorporating new evaluation metrics.

3.3. MEANT for evaluating Chinese

Since UMEANT is shown to be more stable when evaluating translations across different language pairs [36], we use a UMEANT framework along the lines described in [37] for evaluating both English and Chinese.

However, for evaluating Chinese, MEANT has to be equipped with a Chinese shallow semantic parser in order to capture the semantic frames in the Chinese translation output. For this purpose, we used C-ASSERT [38] because of its high accuracy.

Since the primary objective in this experiment is studying the feasibility of tuning MT systems against Chinese MEANT, we limited ourselves to using a window-size-3 context vector model trained on the word segmented monolingual Chinese gigaword corpus, for estimating the phrasal similarity of the semantic role fillers, rather than investigating which combination of window-size, similarity function and phrasal aggregation function that would perform the best in evaluating Chinese.

3.4. Submitted systems

For the English-Chinese TED talks MT task, we submitted translation output from three systems. The primary system is our MEANT-tuned Moses flat phrase-based MT system and the two contrastive systems are the BLEU-tuned Moses flat phrase-based and BLEU-tuned Moses hierarchical phrase-based systems. In this paper, we have also include our latest results on the MEANT-tuned Moses hierarchical phrase-based system.

Table 1: Translation quality of the participated English-Chinese MT systems on the IWSLT 2013 test set where (p) indicates our primary submission; (c1) and (c2) indicate the two contrastive submissions and (n) indicates our not-submitted system.

	char-level		word-level								
	official		official		internal						
System	BLEU	TER	BLEU	TER	BLEU	TER	NIST	WER	PER	CDER	MEANT
(p) MEANT-tuned flat	**18.66**	**70.36**	**10.85**	**78.12**	11.44	79.24	4.25	83.07	64.80	77.04	25.65
(c1) BLEU-tuned flat	18.08	72.00	10.38	82.02	10.93	83.58	4.06	87.19	69.07	81.03	24.88
(c2) BLEU-tuned hier	18.02	72.12	10.37	81.80	10.88	83.63	4.05	87.16	69.44	81.07	23.98
(n) MEANT-tuned hier	—	—	—	—	**11.83**	**72.31**	**4.59**	**76.09**	**58.86**	**70.78**	**25.72**

Table 2: Translation quality of the participated English-Chinese MT systems on the IWSLT 2012 test set where (p) indicates our primary submission; (c1) and (c2) indicate the two contrastive submissions and (n) indicates our not-submitted system.

	word-level (internal)						
System	BLEU	TER	NIST	WER	PER	CDER	MEANT
(p) MEANT-tuned flat	**10.89**	81.20	**4.18**	84.61	67.54	79.38	**24.24**
(c1) BLEU-tuned flat	10.65	86.23	3.98	89.33	72.81	84.17	23.76
(c2) BLEU-tuned hier	10.47	86.53	3.95	89.56	73.34	84.34	22.37
(n) MEANT-tuned hier	9.04	**78.33**	3.73	**81.42**	**66.15**	**76.60**	22.93

For the Chinese-English TED talks MT task, we submitted translation output from four systems. In addition to the primary MEANT-tuned Moses flat phrase-based MT system and the two contrastive BLEU-tuned Moses flat phrase-based and BLEU-tuned Moses hierarchical phrase-based systems, we have also submitted translation output from the contrastive MEANT-tuned Moses hierarchical phrase-based system.

4. Results

Table 1, 2, and 3 show that the MEANT-tuned systems in the English-Chinese TED talks MT task achieves significantly better scores than the two contrastive BLEU-tuned systems across all evaluation metrics on all three test sets. The results is surprising because MEANT-tuned system beats BLEU-tuned systems even on BLEU, the metric which the BLEU-tuned systems are highly optimized on. This encouraging results confirm that MEANT is a better metric for evaluating and tuning MT system on Chinese.

Table 4, 5 and 6 show that the BLEU-tuned systems in the Chinese-English TED talks MT task only performs well on BLEU, the metric which they are highly optimized on. However, MEANT-tuned systems beats the BLEU-tuned systems on other evaluation metrics across all three test sets. More precisely, MEANT-tuned Moses flat phrase-based system achieves the best error metric scores (TER, WER, CDER) while the MEANT-tuned Moses hierarchical phrase-based system achieves better scores in NIST, PER and METEOR. This results show that tuning MT system against BLEU would easily result in overfitting instead of producing good translation in practice. On the other hand, since a high MEANT score rely on correct lexical choices as well as syntactic and semantic structures, tuning MT systems against MEANT would hardly result in overfitting while producing translations that more robustly express the meaning in the original input accurately.

5. Conclusion

In this paper, we have presented the first ever results that tuning a MT system for translating into Chinese against MEANT significantly improves translation quality, instead of tuning against BLEU. MEANT-tuned English-Chinese MT system successfully achieves the best scores across commonly used metrics on all test sets. Since a high MEANT score rely on correct lexical choices as well as syntactic and semantic structures, tuning MT systems against MEANT would hardly result in overfitting while producing translations that more robustly accurately express the meaning in the original input. This effect is more obvious when we are translating into a non-English language.

We have to point out that in this feasibility study we have done minimal adaptation on the settings of MEANT for evaluating Chinese. We expect the performance of

Table 3: Translation quality of the participated English-Chinese MT systems on the IWSLT 2011 test set where (p) indicates our primary submission; (c1) and (c2) indicate the two contrastive submissions and (n) indicates our not-submitted system.

	word-level (internal)						
System	BLEU	TER	NIST	WER	PER	CDER	MEANT
(p) MEANT-tuned flat	**12.24**	79.56	**4.39**	82.42	65.66	77.28	**25.87**
(c1) BLEU-tuned flat	11.12	85.12	4.12	87.94	71.44	82.57	25.33
(c2) BLEU-tuned hier	10.89	83.63	4.05	87.16	69.44	81.07	23.16
(n) MEANT-tuned hier	10.14	**76.66**	3.96	**79.21**	**64.20**	**74.41**	23.51

Table 4: Translation quality of the participated Chinese-English MT systems on the IWSLT 2013 test set where cased and uncased BLEU and TER are the official results. (p) indicates our primary submission; (c1), (c2) and (c3) indicate the three contrastive submissions. MET stands for METEOR.

	cased		uncased									
	official		official		internal							
System	BLEU	TER	BLEU	TER	BLEU	TER	NIST	MET	WER	PER	CDER	MEANT
(p) MEANT-tuned flat	9.58	**74.82**	10.17	**73.75**	10.61	**73.82**	4.57	42.49	**75.66**	58.97	**70.81**	31.42
(c1) MEANT-tuned hier	10.20	75.92	10.79	74.83	11.29	74.59	**4.65**	43.05	77.32	58.96	71.73	**32.50**
(c2) BLEU-tuned flat	10.16	76.05	10.84	74.88	11.32	74.54	**4.65**	43.24	77.05	**58.94**	71.70	31.46
(c3) BLEU-tuned hier	**10.24**	76.95	**10.90**	75.76	**11.41**	75.17	4.62	**43.30**	78.07	59.72	72.39	31.86

MEANT-tuned systems to be even better when the optimal settings are used. This encouraging results show that using MEANT is a promising alternative to BLEU in both evaluating and tuning MT systems to drive the progress of MT research across different languages.

6. Acknowledgment

This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under BOLT contract no. HR0011-12-C-0016, and GALE contract nos. HR0011-06-C-0022 and HR0011-06-C-0023; by the European Union under the FP7 grant agreement no. 287658; and by the Hong Kong Research Grants Council (RGC) research grants GRF620811, GRF621008, and GRF612806. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the EU, or RGC. We are grateful to Pascale Fung, Yongsheng Yang and Zhaojun Wu for sharing the maximum entropy Chinese segmenter and C-ASSERT, the Chinese semantic parser, with us.

7. References

[1] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "BLEU: a method for automatic evaluation of machine translation," in *40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, Pennsylvania, July 2002, pp. 311–318.

[2] Chris Callison-Burch, Miles Osborne, and Philipp Koehn, "Re-evaluating the role of BLEU in machine translation research," in *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 2006, pp. 249–256.

[3] Philipp Koehn and Christof Monz, "Manual and automatic evaluation of machine translation between european languages," in *Workshop on Statistical Machine Translation (WMT-06)*, 2006, pp. 102–121.

[4] Chi-kiu Lo and Dekai Wu, "MEANT: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles," in *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT 2011)*, 2011.

[5] ——, "Unsupervised vs. supervised weight estimation for semantic MT evaluation metrics," in *Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-6)*, 2012.

[6] Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu, "Fully automatic semantic MT evaluation," in *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012.

[7] Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Dan Jurafsky, "Shallow semantic parsing using support vector machines," in *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*, 2004.

Table 5: Translation quality of the participated Chinese-English MT systems on the IWSLT 2012 test set. (p) indicates our primary submission; (c1), (c2) and (c3) indicate the three contrastive submissions.

	uncased (internal)							
System	BLEU	TER	NIST	METEOR	WER	PER	CDER	MEANT
(p) MEANT-tuned flat	10.22	**72.73**	4.47	41.63	**74.42**	59.60	**70.12**	31.94
(c1) MEANT-tuned hier	10.73	73.66	**4.60**	42.00	75.93	**59.32**	70.89	32.29
(c2) BLEU-tuned flat	**10.90**	73.84	4.59	**42.28**	75.80	59.74	71.00	31.45
(c3) BLEU-tuned hier	10.71	73.94	4.56	41.59	76.39	59.76	71.18	**32.60**

Table 6: Translation quality of the participated Chinese-English MT systems on the IWSLT 2011 test set. (p) indicates our primary submission; (c1), (c2) and (c3) indicate the three contrastive submissions.

	uncased (internal)							
System	BLEU	TER	NIST	METEOR	WER	PER	CDER	MEANT
(p) MEANT-tuned flat	11.17	**71.48**	4.66	43.87	**73.05**	58.13	**68.98**	34.12
(c1) MEANT-tuned hierarchical	11.97	71.90	**4.82**	**44.60**	74.16	**57.58**	69.30	**35.47**
(c2) BLEU-tuned flat	11.89	72.30	4.77	44.09	74.43	58.40	69.87	34.39
(c3) BLEU-tuned hierarchical	**12.06**	72.61	4.77	44.16	74.94	58.27	69.89	34.76

[8] George Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *The second international conference on Human Language Technology Research (HLT '02)*, San Diego, California, 2002, pp. 138–145.

[9] Satanjeev Banerjee and Alon Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, Michigan, June 2005, pp. 65–72. [Online]. Available: http://www.aclweb.org/anthology/W/W05/W05-0909

[10] Gregor Leusch, Nicola Ueffing, and Hermann Ney, "CDer: Efficient MT evaluation using block movements," in *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 2006.

[11] Sonja Nießen, Franz Josef Och, Gregor Leusch, and Hermann Ney, "A evaluation tool for machine translation: Fast evaluation for MT research," in *The Second International Conference on Language Resources and Evaluation (LREC 2000)*, 2000.

[12] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul, "A study of translation edit rate with targeted human annotation," in *7th Biennial Conference Association for Machine Translation in the Americas (AMTA 2006)*, Cambridge, Massachusetts, August 2006, pp. 223–231.

[13] Chi-kiu Lo, Karteek Addanki, Markus Saers, and Dekai Wu, "Improving machine translation by training against an automatic semantic frame based evaluation metric," in *51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, 2013.

[14] Chi-kiu Lo and Dekai Wu, "Can informal genres be better translated by tuning on automatic semantic metrics?" in *14th Machine Translation Summit (MT Summit XIV)*, 2013.

[15] Boxing Chen, Roland Kuhn, and George Foster, "Improving AMBER, an MT evaluation metric," in *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012, pp. 59–63.

[16] Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia, "Findings of the 2012 workshop on statistical machine translation," in *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012, pp. 10–51.

[17] Jesús Giménez and Lluís Màrquez, "Linguistic features for automatic evaluation of heterogenous MT systems," in *Second Workshop on Statistical Machine Translation (WMT-07)*, Prague, Czech Republic, June 2007, pp. 256–264. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0738

[18] ——, "A smorgasbord of features for automatic MT evaluation," in *Third Workshop on Statistical Machine Translation (WMT-08)*, Columbus, Ohio, June 2008, pp. 195–198. [Online]. Available: http://www.aclweb.org/anthology/W/W08/W08-0332

[19] Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder, "(meta-) evaluation of machine translation," in *Second Workshop on Statistical Machine Translation (WMT-07)*, 2007, pp. 136–158.

[20] ——, "Further meta-evaluation of machine translation," in *Third Workshop on Statistical Machine Translation (WMT-08)*, 2008, pp. 70–106.

[21] Mengqiu Wang and Christopher D. Manning, "SPEDE: Probabilistic edit distance metrics for MT evaluation," in *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012, pp. 76–83.

[22] Julio Castillo and Paula Estrella, "Semantic textual similarity for MT evaluation," in *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012, pp. 52–58.

[23] Xingyi Song and Trevor Cohn, "Regression and ranking based optimisation for sentence level machine translation evaluation," in *Sixth Workshop on Statistical Machine Translation (WMT 2011)*, 2011, pp. 123–129.

[24] Miguel Rios, Wilker Aziz, and Lucia Specia, "Tine: A metric to assess MT adequacy," in *Sixth Workshop on Statistical Machine Translation (WMT 2011)*, 2011, pp. 116–122.

[25] Maoxi Li, Chengqing Zpng, and Hwee Tou Ng, "Automatic evaluation of Chinese translation output: Word-level or character-level?" in *49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*. Association for Computational Linguistics, 2011, pp. 159–164.

[26] Chang Liu and Hwee Tou Ng, "Character-level machine translation evaluation for language with ambiguous word boundaries," in *50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*. Association for Computational Linguistics, 2012, pp. 921–929.

[27] Chi-kiu Lo and Dekai Wu, "SMT vs. AI redux: How semantic frames evaluate MT more accurately," in *Twenty-second International Joint Conference on Artificial Intelligence (IJCAI-11)*, 2011.

[28] Patrik Lambert, Jesús Giménez, Marta R Costa-jussá, Enrique Amigó, Rafael E Banchs, Lluıs Márquez, and JAR Fonollosa, "Machine translation system development based on human likeness," in *IEEE/ACL 2006 Workshop on Spoken Language Technology (SLT 2006)*. IEEE, 2006, pp. 246–249.

[29] Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng, "Better evaluation metrics lead to better machine translation," in *2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, 2011, pp. 375–384.

[30] Pascale Fung, Grace Ngai, Yongsheng Yang, and Benfeng Chen, "A maximum-entropy Chinese parser augmented by transformation-based learning," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 3, no. 2, pp. 159–168, 2004.

[31] Franz Josef Och, "Minimum error rate training in statistical machine translation," in *41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, Sapporo, Japan, July 2003, pp. 160–167. [Online]. Available: http://www.aclweb.org/anthology/P03-1021

[32] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst, "Moses: Open source toolkit for statistical machine translation," in *Interactive Poster and Demonstration Sessions of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, Prague, Czech Republic, June 2007, pp. 177–180.

[33] David Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007. [Online]. Available: http://aclweb.org/anthology-new/J/J07/J07-2003.pdf

[34] Andreas Stolcke, "SRILM – an extensible language modeling toolkit," in *7th International Conference on Spoken Language Processing (ICSLP2002 - INTERSPEECH 2002)*, Denver, Colorado, September 2002, pp. 901–904.

[35] Omar F. Zaidan, "Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems," *The Prague Bulletin of Mathematical Linguistics*, vol. 91, pp. 79–88, 2009.

[36] Matouš Macháček and Ondřej Bojar, in *8th Workshop on Statistical Machine Translation (WMT 2013)*. Association for Computational Linguistics, 2012, pp. 921–929.

[37] Chi-kiu Lo and Dekai Wu, "MEANT at WMT 2013: A tunable, accurate yet inexpensive semantic frame based mt evaluation metric," in *8th Workshop on Statistical Machine Translation (WMT 2013)*, 2013.

[38] Zhaojun Wu, Yongsheng Yang, and Pascale Fung. (2006) C-ASSERT: Chinese shallow semantic parser. Website. HKUST. [Online]. Available: http://hlt030.cse.ust.hk/research/c-assert/

The NICT ASR System for IWSLT 2013

Chien-Lin Huang, Paul R. Dixon, Shigeki Matsuda, Youzheng Wu, Xugang Lu, Masahiro Saiko, Chiori Hori

Spoken Language Communication Laboratory
National Institute of Information and Communications Technology, Kyoto, Japan
chien-lin.huang@nict.go.jp

Abstract

This study presents the NICT automatic speech recognition (ASR) system submitted for the IWSLT 2013 ASR evaluation. We apply two types of acoustic features and three types of acoustic models to the NICT ASR system. Our system is comprised of six subsystems with different acoustic features and models. This study reports the individual results and fusion of systems and highlights the improvements made by our proposed methods that include the automatic segmentation of audio data, language model adaptation, speaker adaptive training of deep neural network models, and the NICT SprinTra decoder. Our experimental results indicated that our proposed methods offer good performance improvements on lecture speech recognition tasks. Our results denoted a 13.5% word error rate on the IWSLT 2013 ASR English test data set.

1. Introduction

The IWSLT 2013 Automatic Speech Recognition is an ongoing evaluation whose goal is to automatically transcribe TED [1] talks from audio to text [1]. TED is a nonprofit organization that promotes the dissemination of ideas. People can access TED talks on its website. Due to speech disfluency, emotional speech, noisy speech, different channels and speakers, the automatic transcription of TED talks is challenging. This year, the evaluation contains English and German speech materials as well as the automatic and mandatory segmentation of audio data. Since some talks are with non-native speakers, this year's evaluations are particularly challenging.

Automatic speech recognition has been widely applied in different kinds of applications [2]-[4]. To achieve better speech recognition performance, many techniques [5]-[9] have been proposed to address the problems in speech recognition. Cui et al. [5] presented a new semi-supervised learning method that exploits cross-view transfer learning for speech recognition through a committee machine that consists of multiple views learned from different acoustic features and randomized decision trees. A multi-objective scheme is generalized to a unified semi-supervised learning framework that can be interpreted into a variety of learning strategies under different weighting schemes. Huang et al. [6] proposed a joint analysis approach which simultaneously considers the vocal tract length normalization and the averaged temporal information of cepstral features. The Gaussian mixture model estimates conditional parameters in a data-driven manner. Chelba et al. [8] reviewed an approach to acoustic modeling that borrows from n-gram language modeling to increase both the amount of training data and the model size to approximately 100 times larger than the current sizes used in ASR. They experimented with contexts that span seven or more context-independent phones, and up to 620 mixture components per state. Hinton et al. [9] provided an overview of deep neural networks (DNNs) for acoustic modeling. Most speech recognition systems use hidden Markov models (HMMs) to deal with the temporal variability of speech and Gaussian mixture models (GMMs) to determine how well each state of each HMM fits a frame or a short window of frames of coefficients that represents the acoustic input. DNNs trained using new methods have outperformed GMMs on a variety of speech recognition benchmarks. In addition, Kaldi[2] [10] is an open-source toolkit of ASR written in C++. The core library support state-of-the-art techniques of modeling and feature extraction including DNN models, subspace Gaussian mixture models (SGMMs), decoder of finite-state transducers, and so on. In this study, we adopt Kaldi and NICT SprinTra for ASR system development and investigate speech recognition techniques on data analysis, feature extraction, acoustic and language models, and speech decoders.

The rest of this paper is organized as follows. Section 2 introduces data analysis and segmentation. We present the construction of combining multiple features and models for lecture speech recognition in Section 3. In Section 4, we describe our experiment setup, experiment results as well as a discussion of the results. Finally, we conclude this work in Section 5.

2. Data Analysis and Segmentation

We used three types of speech data to build acoustic models: the Wall Street Journal (WSJ), HUB4 English Broadcast news, and collected TED talks. We obtained WSJ and HUB4 from the Linguistic Data Consortium (LDC[3]). We crawled 760 TED talks from its online website published before December 31, 2010. The data are summarized in Table 1. WSJ is read speech. HUB4 is spontaneous broadcast news speech. TED is lecture style speech. Totally, we have about 300 hours of speech to build acoustic models with transcripts.

Both WSJ and HUB4 provide manual transcripts that can be directly used for acoustic model training. Text captions or subtitles of TED are provided with the speech recording, but speech segmentation and word alignment are not available. We used the SailAlign toolkit for speech segmentation and speech-text alignment [11]. SailAlign, which provides decoder-based segmentation with acoustic and language model adaptation, runs with HTK in which the acoustic model is trained by WSJ. Based on the segmentation results, the

[1] http://www.ted.com/

[2] http://kaldi.sourceforge.net/

[3] http://www.ldc.uppen.edu/

Table 1: Details of acoustic training data.

Name	Data	Type	Hours
TED	-	Lecture	167.8
HUB4	LDC97S44, LDC98S71	Broadcast	62.9
WSJ	LDC93S6B, LDC94S13B	Read	81.1

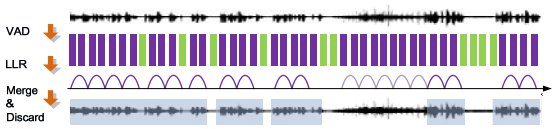


Figure 1: Illustration of the automatic segmentation of audio data.

speech-text alignment can be viewed as text-text alignment using dynamic programming to minimize the distance between reference and hypothesized texts.

In this study, the techniques of speaker clustering and automatic segmentation are applied to training and test audio data sets. First, speaker clustering has been widely adopted for clustering speech data based on speaker characteristics so that speaker-based cepstral mean normalization (CMN) and speaker adaptive training (SAT) [12] can be applied for better automatic speech recognition performance. TED talks are not always monologue; they might include interviews or conversations. We apply the vector space strategy to represent spoken utterances and conduct speaker clustering to group the spoken utterances into a number of speaker clusters in each talk. Experimental analysis is available in our earlier study [13].

Second, the length of a TED talk may range from 3 to 18 minutes with speech, laugh, applause, music, etc. For a good speech transcription, we apply the automatic segmentation processing to the audio data to remove non-speech segments (Fig. 1). Energy-based voice activity detection (VAD) is first used to detect the voice segments. Then the log-likelihood score with sliding windows is computed to detect speech/non-speech segments based on two GMMs trained using labeled speech/non-speech data. Finally, we merge the speech segments with a short interval between them and discard short segments. Merging and discard are based on a threshold of 170 ms.

3. System Description

3.1. Feature Extraction

Feature extraction is crucial to estimate numerical representation from speech samples. In this study, we extracted two sets of acoustic features to build acoustic models. The first set is Mel-frequency cepstral coefficient (MFCC), which is popular in speech recognition applications [14]. In MFCC feature extraction, 16-KHz speech input is coded with 13-demensional MFCCs with a 25ms window and a 10ms frame-shift. Each frame of the speech data is represented by a 39-dimensional feature vector that consists of 13 MFCCs with their deltas and double-deltas. Nine consecutive feature frames are spliced and projected to 40 dimensions using linear discriminant analysis (LDA) and maximum likelihood linear transformation (MLLT). The second acoustic feature is a perceptual linear predictive cepstrum (PLP) [15], which has the same LDA and MLLT. Both have 40 dimensions.

3.2. Subsystem Descriptions

The HMM models were with maximum 10,000 tied states and 160,000 Gaussian mixture components. We investigated three kinds of acoustic models: training of maximum mutual information (MMI), SGMM, and DNN.

Maximum Mutual Information Training: We maximized the auxiliary function in the M-steps of the EM estimation of the HMM parameters. The likelihood of the data given HMM is bound to increase when the value of the auxiliary function increases. In model space MMI training, we maximize a model's correctness by formulating an objective function and penalizing confusable models to the true model [16]. fMMI is feature space discriminative training with the same objective function as model space MMI training. After applying a global matrix, a high dimension feature vector is projected and added to the original features. In this study, we first apply speaker adaptive training on a triphone HMM system. Then discriminative training is applied with a feature space boosted fMMI followed by tree rebuilding and model space MMI training with indirect differential [17].

Subspace GMM Training: The subspace Gaussian mixture model is a compact representation of a large collection of a mixture of Gaussian models [18]-[20]. SGMM's basic idea is that all phonetic states share a common GMM structure, but the means and mixture weights vary in the total parameter space. Since most parameters are shared, we have more robust parameter estimation. We initialize the model by training a single GMM on all the speech classes that are pooled together. This is the universal background model (UBM). We use a total of 800 Gaussians in the UBM. Before SGMM training, SAT is used on the triphone system that is related to MLLR adaptation.

DNN Training: The deep neural networks are feed-forward, artificial neural networks that show more than one hidden layers between inputs and outputs [9, 21]. Recently, DNN has become a popular technique because it indicates good results for modeling speech acoustics. Many studies show that neural network based HMMs significantly outperform Gaussian mixture model based HMMs. In this study, starting from a DNN trained using cross-entropy, sequence discriminative training is then applied based on the state level minimum Bayesian risk criterion (sMBR) [22]. sMBR's objective function is explicitly designed to minimize the expected error corresponding to state labels, but we minimize the cross-entropy at the frame-level. We build DNNs by using five hidden layers and 2100 neurons (the structure is 300-2100-2100-2100-2100-2100-8070) (Fig. 2). DNN's input features are obtained by splicing together 15 frames (seven on each side of the current frame) and projected down to 300 dimensions using LDA. To better fit new speakers and environments, DNN acoustic models have been further adapted for specific talks using speaker adaptive training. Due to the limited amount of data in each talk, an efficient and effective method of speaker adaptive training of DNN models is only to adapt the middle layer (the third hidden layer). Speaker adaptation for DNN is difficult. In

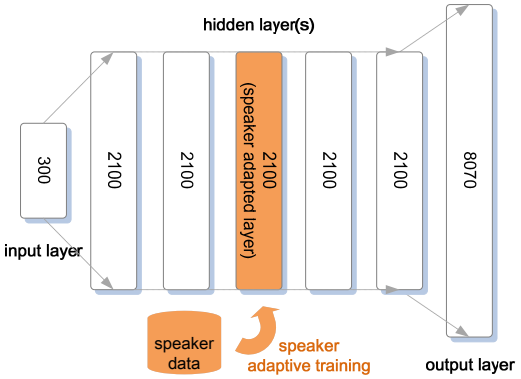


Figure 2: Illustration of speaker adaptive training of deep neural network models.

most studies, a speaker independent DNN (SI-DNN) is first trained. Then a speaker adaptation DNN is done by retraining the DNN parameters for different speakers either on all layers or some specific layers in the DNN [23, 24].

In this study, we propose a new speaker adaptive DNN training framework (SAT-DNN). We first assume that speaker specific processing is done in one layer in the DNN. All other layers are related to the speaker independent processing. Based on this assumption, we constructed a DNN with one layer as a speaker dependent layer, and the other layers are shared cross all speakers. In the DNN training, the parameters related to the speaker dependent layer are modified for each speaker while the parameters for all the shared layers are updated for all speakers. Explicitly specifying one layer as a speaker dependent layer in training focuses the training much more on speaker adaptation in DNN.

3.3. N-best ROVER

We considered a combination of two subsystems of MMI and SGMM in last year's evaluation [25]. This year, we built six subsystems using three types of acoustic models with two types of acoustic features. We integrated multiple complementary features and models for a better performance (Fig. 3). Several methods can be used to combine different recognition results. One popular approach is called recognizer output voting error reduction (ROVER) [26, 27]. Cui et al. [5] applied ROVER as a decision committee that votes for the labels of unlabeled data by cross validation. The combination can be carried out at the text output level as an n-best ROVER by output voting. We combine all decoding directories by composing the lattices. In this paper, different combination weights are applied to MMI, SGMM and DNN subsystems with 0.25, 0.25, and 0.5, respectively.

3.4. Language Model Adaptation and RNN Rescoring

We used the CMU pronouncing dictionary which has 133.3K words. We extended 39 phones of the dictionary to a 336 monophone set based on the accent and position information. The language models (LM) are modified Kneser-Ney smoothed 4-gram LMs trained on official data using the SRILM toolkit [28]. We used two different pruning 4-gram

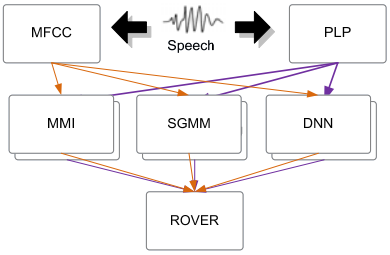


Figure 3: The combination of multiple systems for speech recognition using ROVER.

LMs in our experiments. The small 4-gram LM has 212 MB, and the big 4-gram LM has 9.6 GB and its perplexity is 115.4. Due to hardware and software limits, speech is decoded using the small 4-gram LM and rescored using the big 4-gram LM on MMI and SGMM subsystems. We use the first pass decoding results to adapt the language models that are used for second pass decoding [29]. In addition to conventional 4-gram LMs, we also applied a recurrent neural network (RNN) based LM [30] to rescore the n-best results. The sigmoidal recurrent network was built with the RNN-LM toolkit [31].

3.5. NICT SprinTra Decoder

In this paper, the ASR decoding process was based on weighted finite state transducers (WFSTs) [32], which integrate the acoustic and language models at the lattice level. We used the NICT SprinTra decoder, which has two major advantages [33]. First, the NICT SprinTra has smaller memory requirement and shows much faster decoding speed than the Kaldi decoder. Both NICT SprinTra and Kaldi use OpenFST [4] tools and library [34], but we use different structures to build the decoding graph. We also computed the so-called real-time (RT) factor. On the small 4-gram LM, SprinTra's decoding time was about 0.729×RT measured on an Intel Xeon CPU at 2.6GHz. This is better than the 1.023×RT of Kaldi and about a 30% difference in decoding time. Running on the big 4-gram LM, NICT SprinTra is ten times faster than Kaldi. Second, since the NICT SprinTra decoder decodes speech using the one pass method without language model rescoring, it is more accurate than decoding using language model rescoring. The word error rates vary from 0.1% to 0.3% between NICT SprinTra and Kali. This also denotes the gain using the big 4-gram LM decoding or rescoring.

4. Experiments

4.1. Training of Different Acoustic Data Sets

We experimented on the IWSLT 2013 ASR English test data set, which contained 4.5 hours of lecture speech, with 28 talks including 14 males and 14 females. There were at least eight non-native speakers (four males and four females) and one child. The effect of reverberation can be found in ten lectures. Non-native speech may be the main reason for the decrease of recognition accuracy. System performance was assessed using Word Error Rate (WER). Table 2 shows the results of

[4] http://www.openfst.org/

Table 2: MFCC-DNN subsystem results on training of different acoustic data sets.

Data	TED	TED+HUB4	TED+HUB4+WSJ
WER	16.9%	16.1%	15.7%

Table 3: Improvements by adding of different techniques on the IWSLT ASR 2013 English test data set.

System	WER	Reduction
MFCC-DNN baseline	15.7%	-
+ Six ROVER subsystems	14.8%	5.7%
+ Automatic segmentation	14.3%	3.4%
+ LM adaptation	14.1%	1.4%
+ SAT on DNN	13.5%	4.3%

the MFCC-DNN subsystem using different training data sets. All results were conducted on the entire lecture without any segmentation. Our experiments indicated that more data improved performance. Only the TED training was not good enough to recognize the TED speech of the IWSLT 2013 ASR English test data set. We achieved 15.7% WER using TED+HUB4+WSJ for the single MFCC-DNN subsystem, although HUB4 and WSJ were different types of speech from TED. We used the 15.7% WER result as the baseline in the following experiments.

4.2. Step-by-Step Improvements

Based on an MFCC-DNN baseline of 15.7% WER, Table 3 summarizes the step-by-step WER reductions with our proposed methods. First, the WER can be reduced to 14.8% using six ROVER subsystems. Due to error propagation and non-speech segments, the entire lecture decoding indicated poor performance. Adding an automatic segmentation technique reduced the WER from 14.8% to 14.5%, or 3.4% relative WER reduction. In addition, WER reductions of 1.4% and 4.3% were achieved for LM adaptation and SAT on DNN. Both adaptation methods were used to adjust models to better fit new speakers and environments. Our proposed methods offered more than 10% WER reduction on average. Our best result was 13.5% WER on the IWSLT 2013 ASR English test data set. Note that the application order of these techniques impacted the gain. For example, to get good speech transcriptions for adaptation, the LM adaptation technique is based on automatic segmentation results of audio data and six ROVER subsystems. In addition, the single MFCC-DNN subsystem indicated about 1.0% absolute WER reduction using the automatic segmentation of audio data, LM adaptation, and SAT on DNN.

4.3. Subsystems and ROVER Results

Table 4 shows the speech recognition evaluation of a combination of multiple features and models. Our experiments suggest the following observations. First, the MFCC and PLP features indicated similar results in most cases. Second, we evaluated the results of individual subsystems (1S). The DNN acoustic models significantly outperformed SGMM and MMI. Even the SGMM and MMI performances were much worse than DNN, and a combination of six subsystems (6S) further reduced the WER using ROVER. Compared with the 13.5% WER of six ROVER subsystems, the best result of the single MFCC-DNN system was 14.0% WER. The ROVER result was about 13.9% if we only considered MFCC features on three acoustic models (3S). Interestingly, we can obtain 13.9% WER using ROVERs of MFCC-DNN and PLP-DNN.

Table 4: Results (in %WER) of different subsystems and ROVER on the IWSLT ASR 2013 English test data set.

Feature	Model	1S	3S	6S
MFCC	MMI	19.7%	13.9%	13.5%
	SGMM	20.4%		
	DNN	14.0%		
PLP	MMI	20.2%	14.0%	
	SGMM	20.6%		
	DNN	14.1%		

4.4. Summary Results

Table 5 indicated the detailed results of each talk on the IWSLT ASR 2013 English test data set. Non-native speakers have the higher error rate in most cases. The WER is lower than 5% in the best condition but over 30% in the worst condition. Due to child voices and non-native speakers, talkid1699 denoted the worst recognition result. Furthermore, the IWSLT ASR 2011 (tst2011) and 2012 (tst2012) test data sets were used as progressive tests. There are eight and 11 talks in tst2011 and tst2012, respectively. Compared with this year's result of 13.5% WER, 7.7% and 8.2% WER results were achieved for tst2011 and tst2012 using our proposed approaches.

5. Conclusions

In this study, we propose a combination of multiple features and models for lecture speech recognition. We build six subsystems using three types of acoustic models (MMI, SGMM, and DNN) with two types of acoustic features (MFCC and PLP). The n-best ROVER denotes a good solution for a subsystem combination. We discover techniques of discriminative training and the adaptation of both acoustic and language models show great contributions to ASR. We propose the automatic segmentation of audio data, language model adaptation, speaker adaptive training of DNN models, and NICT SprinTra decoder. The results of our proposed methods demonstrate good performance improvement on the IWSLT 2013 ASR data set. There is still room for improvement when considering both good and a large amount of data.

6. References

[1] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 10th IWSLT Evaluation Campaign," *International Workshop on Spoken Language Translation (IWSLT)*, 2013.

Table 5: Detailed results of each talk on the IWSLT ASR 2013 English test data set.

Speaker	Gender	Dialect	# Second	# Sentence	# Word	% Corr	% Sub	% Del	% Ins	% Err	% S.Err
talkid1518	male	non-native, Deutsch	588	112	1247	89.4	8.5	2.1	1.7	12.3	53.6
talkid1520	male	south Asian	640	123	1416	88.7	8.1	3.2	2.1	13.4	52.0
talkid1532	female	native	569	113	1529	89.3	7.6	3.1	2.2	12.9	69.0
talkid1534	male	native	343	66	1165	81.6	9.2	9.2	0.9	19.3	84.8
talkid1539	male	African American	250	31	546	92.9	5.7	1.5	2.7	9.9	64.5
talkid1541	female	native	1141	247	2567	96.6	2.7	0.7	0.8	4.1	26.7
talkid1548	male	native	381	75	1083	88.8	7.5	3.7	1.9	13.1	62.7
talkid1553	female	native	359	79	804	96.3	2.6	1.1	0.6	4.4	31.6
talkid1592	female	native	251	39	670	98.4	1.2	0.4	0.7	2.4	30.8
talkid1600	female	African American	301	79	702	90.2	6.6	3.3	1.6	11.4	46.8
talkid1610	male	Italian	382	67	949	94.4	4.0	1.6	0.9	6.5	52.2
talkid1617	male	native	1009	172	2221	87.4	10.4	2.2	2.3	14.9	56.4
talkid1634	male	native	199	50	550	94.5	2.5	2.9	0.2	5.6	36.0
talkid1637	female	native	671	166	2021	96.0	2.5	1.5	0.4	4.4	34.9
talkid1640	male	native	544	108	1632	89.5	5.8	4.7	0.4	10.9	52.8
talkid1646	female	African American	508	80	927	83.7	13.7	2.6	2.5	18.8	71.3
talkid1647	female	native	558	106	1796	90.4	5.9	3.7	1.8	11.4	57.5
talkid1649	male	native	1047	247	3250	84.6	8.1	7.3	0.9	16.3	62.3
talkid1651	female	native	693	187	1739	95.3	2.7	2.0	0.7	5.4	29.9
talkid1654	female	native	940	244	2284	97.4	1.7	0.9	0.4	3.0	20.1
talkid1658	female	native	1077	209	2997	89.0	4.9	6.1	1.1	12.1	57.4
talkid1659	female	non-native, Egypt	562	112	1017	91.4	6.8	1.8	3.1	11.7	48.2
talkid1665	male	non-native, Deutsch	391	69	896	79.2	15.2	5.6	2.5	23.2	75.4
talkid1666	female	non-native, Afghanistan	554	150	1049	95.7	2.7	1.6	0.9	5.1	28.0
talkid1673	male	native	1072	235	3430	79.7	12.0	8.3	1.6	21.9	72.3
talkid1685	male	African American	614	139	1623	72.8	17.7	9.5	1.6	28.8	82.0
talkid1694	female	north Korean	715	115	1617	74.3	21.6	4.1	6.5	32.2	87.8
talkid1699	male	Kenya child	425	126	1021	64.7	26.0	9.3	2.4	37.6	78.6

[2] J.-R. Ding, C.-L. Huang, J.-K. Lin, J.-F. Yang, and C.-H. Wu, "Interactive Multimedia Mirror System Design," *IEEE Trans. Consumer Electronics*, vol. 54, no. 3, pp. 972–980, 2008.

[3] C.-L. Huang and C.-H. Wu, "Spoken Document Retrieval Using Multi-Level Knowledge and Semantic Verification," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2551–2560, 2007.

[4] C.-H. Wu, C.-H. Hsieh, and C.-L. Huang, "Speech Sentence Compression Based on Speech Segment Extraction and Concatenation," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 434–438, 2007.

[5] X. Cui, J. Huang, and J.-T. Chien, "Multi-View and Multi-Objective Semi-Supervised Learning for HMM-Based Automatic Speech Recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 447–460, 2012.

[6] C.-L. Huang, C. Hori, H. Kashioka, and B. Ma, "Joint Analysis of Vocal Tract Length and Temporal Information for Robust Speech Recognition," in *Proc. of ICASSP*, 2013.

[7] C.-L. Huang and C.-H. Wu, "Generation of Phonetic Units for Mixed-Language Speech Recognition Based on Acoustic and Contextual Analysis," *IEEE Trans. Computers*, vol. 56, no. 9, pp. 1225–1233, 2007.

[8] C. Chelba, P. Xu, F. Pereira, and T. Richardson, "Large Scale Distributed Acoustic Modeling with Back-Off N-Grams," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 21, no. 6, pp. 1158–1169, 2013.

[9] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, pp. 82–97, 2012.

[10] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Proc. of ASRU*, 2011.

[11] A. Katsamanis, M. Black, P. Georgiou, L. Goldstein and S. Narayanan, "SailAlign: Robust Long Speech-Text Alignment," in *Proc. of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, 2011.

[12] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A Compact Model for Speaker-Adaptive Training," in *Proc. International Conference on Spoken Language Processing (ICSLP)*, vol. 2, pp. 1137–1140, 1996.

[13] C.-L. Huang, C. Hori, H. Kashioka, and B. Ma, "Speaker Clustering Using Vector Representation with Long-Term

Feature for Lecture Speech Recognition," in *Proc. of ICASSP*, 2013.
[14] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, N.J.: Prentice Hall, 1993.
[15] H. Hermansky, "Perceptual Linear Predictive (PLP) analysis of speech," *Journal of the Acoustic Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.
[16] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. of ICASSP*, 2008.
[17] D. Povey, S. M. Chu, J. Pelecanos, and H. Soltau, "Approaches to Speech Recognition based on Speaker Recognition Techniques," chapter in forthcoming GALE book.
[18] X. Zhang, K. Demuynck, D.V. Compernolle, and H.V. Hamme, "Subspace-GMM Acoustic Models for Under-Resourced Languages: Feasibility Study," in *Proc. of SLTU*, 2012.
[19] L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N.K. Goel, M. Karafiat, A. Rastrow, R.C. Rose, P. Schwarz, and S. Thomas, "Subspace Gaussian Mixture Models for Speech Recognition," in *Proc. of ICASSP*, 2010.
[20] N.T. Vu, T. Schultz, and D. Povey, "Modeling gender dependency in the Subspace GMM framework," in *Proc. of ICASSP*, 2012.
[21] A. K. Jain and J. Mao, "Artificial Neural Networks: A Tutorial," *IEEE Computer*, vol. 29, no. 3, pp. 31–44, 1996.
[22] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. of Interspeech*, 2013.
[23] H. Liao, "Speaker Adaptation of context dependent deep neural networks," in *Proc. of ICASSP*, 2013.
[24] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-Divergence Regularized Deep Neural Network Adaptation For Improved Large Vocabulary Speech Recognition," in *Proc. of ICASSP*, 2013.
[25] H. Yamamoto, Y. Wu, C.-L. Huang, X. Lu, P. R. Dixon, S. Matsuda, C. Hori, and H. Kashioka, "The NICT ASR System for IWSLT2012," in *Proc. of IWSLT*, 2012.
[26] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proc. IEEE Workshop Automatic Speech Recognition Understanding*, pp. 347–354, 1997.
[27] X. Cui, J. Xue, B. Xiang, and B. Zhou, "A study of bootstrapping with multiple acoustic features for improved automatic speech recognition," in *Proc. of Interspeech*, pp. 240–243, 2009.
[28] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," in *Proc. of ICSLP*, vol. 2, pp. 901–904, 2002.
[29] Y. Wu, K. Abe, P.R. Dixon, C. Hori, and H. Kashioka, "Leveraging Social Annotation for Topic Language Model Adaptation," in *Proc. of Interspeech*, 2012.
[30] T. Mikolov, M. Karafiát, L. Burget, J. ˇCernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. of Interspeech*, 2010.
[31] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Černocký, "RNNLM - Recurrent Neural Network Language Modeling Toolkit," in *Proc. of ASRU*, 2011.
[32] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech and Language*, vol. 20, no. 1, pp. 69–88, 2002.
[33] P.R. Dixon, C. Hori, and H. Kashioka, "Development of the SprinTra WFST Speech Decoder," *NICT Research Journal*, pp 15-20, 2012
[34] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: a general and efficient weighted finite-state transducer library," in *Proc. of ICAA*, pp. 11–23, 2007.

FBK @ IWSLT 2013 - ASR tracks

D. Falavigna, R. Gretter, F. Brugnara, D. Giuliani, R. H. Serizel

HLT research unit, FBK, 38123 Povo (TN), Italy

(falavi,gretter,brugnara,giuliani,serizel)@fbk.eu

Abstract

This paper reports on the participation of FBK at the IWSLT2013 evaluation campaign on automatic speech recognition (ASR): precisely on both English and German ASR track. Only primary submissions have been sent for evaluation.

For English, the ASR system features acoustic models trained on a portion of the TED talk recordings that was automatically selected according to the fidelity of the provided transcriptions. Two decoding steps are performed interleaved by acoustic feature normalization and acoustic model adaptation. A final step combines the outputs obtained after having rescored the word graphs generated in the second decoding step with 4 different language models. The latter are trained on: out-of-domain text data, in-domain data and several sets of automatically selected data.

For German, acoustic models have been trained on automatically selected portions of a broadcast news corpus, called "Euronews". Differently from English, in this case only two decoding steps are carried out without making use of any rescoring procedure.

1. Introduction

The IWSLT 2013 Evaluation Campaign, similarly to the one carried out for IWSLT2012 [1], addresses the automatic transcription/translation of TED Talks [1]: a collection of public speeches on a variety of topics.

This year, for the transcription of English audio tracks we have focused on automatic selection and exploitation of training data, both audio and text.

We have trained acoustic models (AMs) on both in-domain audio data, extracted from videos downloaded from TED talk WEB site (i.e. http:// www.ted.com/talk/), and out-of-domain data including the broadcast news speech corpus "HUB4" provided by linguistic data consortium (LDC). Since audio recordings of TED talks have only associated "non-exact" transcriptions, a lightly supervised training approach [2] has been applied in order to select reliable data for AM training.

For language model (LM) training, out-of-domain data come from several sources and contain about 5 billions (5G) of words. In addition, a set of in-domain data, containing about 2.7 millions (2.7M) of words, has been provided by organizers. Then, similarly to what done in our ASR submission of last year [3], we have used the automatic transcription of each given English TED talk for automatically selecting from the out-of-domain text data a set of 100M words. From each text source a corresponding LM was trained and used for rescoring word graphs (WGs) generated in the second decoding step. In addition, an interpolated LM, resulting from the linear interpolation of all of the different LMs, has been used for rescoring. Our primary submission has been obtained after having combined, using the ROVER approach [4], all of the different rescored ASR hypotheses.

German AMs were trained using "Euronews" videos downloaded in the last few years from the portal http://de.euronews.com/. Since each video has associated a reference text, that doesn't not contain the exact transcription of the corresponding audio track, we have applied also in this case a lightly supervised approach [2] for AM training. Doing this, about 256 hours of audio data, including silences, were selected. Cut-off date for the latter data was March 2013.

German data for LM has been first normalized applying a procedure that split numbers and compound words automatically found inside training documents. One 4-gram LM has been trained on about 1.7G of words, coming from news, European Parliament, IWSLT13 training data. Cut-off of training data date was end of June 2012.

2. Automatic transcription systems

In this section we summarize the main features of the FBK primary systems used for transcribing TED talks delivered in English and German. This year, differently from previous evaluation campaigns, time boundaries of speech segments to be transcribed are not given. Hence, automatic speech segmentation has to be carried out.

2.1. Automatic speech segmentation

The input audio signal is first divided into segments by a start-end point detector module. The obtained segmentation is refined using an acoustic classifier, based on Gaussian mixture models (GMMs), which also performs classification of segments into several classes including non-speech classes [5]. Then, the obtained homogeneous non-overlapping speech segments are clustered by using a method based on the Bayesian information criterion. At the end of this process, each audio file to transcribe has assigned a set of temporal segments, each having associated a label that indicates the cluster to which it belongs (e.g. "female_1", "male_1", etc). The resulting segmentation and clustering is then exploited by the recognition system to perform cluster-wise feature normalization and acoustic models adaptation

[1] http://www.ted.com/talks

during two decoding passes described below.

3. English transcription system

3.1. Acoustic data selection

For AM training, HUB4 speech corpus was initially used. It contains around 164 hours of broadcast news speech with related word transcriptions, that include also "filler-words". These latter ones have been mapped into 6 different "spontaneous speech" models. After having trained triphone Hidden Markov Models (HMMs) on HUB4, domain specific acoustic data (i.e. a certain number of TED talks recordings) were exploited for lightly supervised training [2].

Recordings of TED talks released before the cut-off date, 31 December 2010, were downloaded with the corresponding subtitles which are content-only transcriptions of the speech. In content-only transcriptions anything irrelevant to the content is ignored, including most non-verbal sounds, false starts, repetitions, incomplete or revised sentences and superfluous speech by the speaker. A simple but robust procedure was implemented to select only audio data with an accurate transcription.

The collected data consisted in 820 talks, for a total duration of $\sim$216 hours, with $\sim$166 hours of actual speech. The provided subtitles are not a verbatim transcription of the speeches, hence the following procedure was applied to extract segments that can be deemed reliable. The approach is that of selecting only those portions in which the human transcription and an automatic transcription agree. To this end, a "background" 4-gram language model was first trained on all the talk transcriptions. Subsequently, a specific Language Model (LM) was built for each talk by adapting the language model to the human transcription of the talk. A preliminary automatic transcription was performed on the talks with the pre-trained HUB4 AM and the talk-specific LM (note that, in doing this, optional "spontaneous speech" models were allowed among words). The output of the system was aligned with the reference transcriptions, and the matching segments were selected, resulting in an overlap of $\sim$120 hours of actual speech out of the total of 166. By using these segments together with the segments labeled as silence, a TED-specific acoustic model was trained, as detailed in the following section. The label/select/train procedure was repeated two more times, resulting in a portion of selected actual speech that grew to $\sim$142 hours and then to $\sim$144 hours. Given the modest improvement in the third iteration, the procedure was not repeated further. In conclusion, the method made available 87% of the training speech, which was considered satisfactory.

In total, after automatic selection, we get around 307 hours ($\sim$164 hours from HUB4 plus $\sim$144 hours from TED recordings) of transcribed speech for training acoustic models.

3.2. AM training

Thirteen Mel-frequency cepstral coefficients, including the zero order coefficient, are computed every 10ms using a Hamming window of 20ms length. First, second and third order time derivatives are computed after segment-based cepstral mean subtraction to form 52-dimensional feature vectors. Acoustic features are normalized and HLDA-projected to obtain 39-dimensional feature vectors as described below.

AMs were trained exploiting a variant of the speaker adaptive training method based on Constrained Maximum Likelihood Linear Regression (CMLLR) [6]. In our training variant [7, 8, 9] there are two sets of AMs: the target models and the recognition models. The training procedure makes use of an affine transformation to normalize acoustic features on a cluster by cluster basis with respect to the target models. For each cluster of speech segments, an affine transformation is estimated through CMLLR [6] with the aim of minimizing the mismatch between the cluster data and the target models. Once estimated, the affine transformation is applied to cluster data in order to normalize acoustic features with respect to the target models. Recognition models are then trained on the normalized data. Leveraging on the possibility that the structure of the target and recognition models can be determined independently, a Gaussian Mixture Model (GMM) can be adopted as the target model for training AMs used in the first decoding pass [7]. This has the advantage that, at recognition time, word transcriptions of test utterances are not required for estimating feature transformations. Instead, target models for training recognition models used in a second or third decoding pass are usually triphones with a single Gaussian per state [8]. In all cases, the same target models are used for estimating cluster-specific transformations during training and recognition.

In the current version of the system, a projection of the acoustic feature space based on Heteroscedastic Linear Discriminant Analysis (HLDA) is embedded in the feature extraction process as follows. A GMM with 1024 Gaussian components is first trained on an extended acoustic feature set consisting of static acoustic features plus their first, second and third order time derivatives. For each cluster of speech segments, a CMLLR transformation is then estimated w.r.t. the GMM and applied to acoustic observations. After normalizing the training data, an HLDA transformation is estimated w.r.t. a set of state-tied, cross-word, gender-independent triphone HMMs with a single Gaussian per state, trained on the extended set of normalized features. The HLDA transformation is then applied to project the extended set of normalized features in a lower dimensional feature space, that is a 39-dimensional feature space. Recognition models used in both the first and second decoding pass are trained from scratch on normalized HLDA-projected features. HMMs for the first decoding pass are trained through a conventional maximum likelihood procedure. Recognition models used in the second decoding pass are speaker-adaptively trained, exploiting as target-models triphone HMMs with a single Gaussian density per state.

For each phone set and decoding pass, a set of state-tied, cross-word, gender-independent triphone HMMs were trained for recognition. Around 170,000 Gaussian densities, with diagonal covariance matrices, were allocated for each model set.

3.3. LM training

Text data used for training LMs are those released for the IWSLT2013-SLT Evaluation Campaign. Before training LMs, texts were cleaned, normalized (punctuation was removed, numbers and dates were expanded) and double lines were removed. Then, they have been grouped into the following three sets, on which a corresponding LM was trained:

- **giga5** GIGAWORD 5-th edition. Contains documents stemming from seven distinct international sources of English newswire. It is released from the Linguistic Data Consortium (see http://www.ldc.upenn.edu/). In total it contains about 4G words.
- **wmt13** Formed by documents in WMT12 news crawl, news commentary v7 and Europarl v7 (see IWSLT2013 official web site for some more details about these corpora). In total it contains about 1G words.
- **ted13** An in-domain set of texts extracted from TED talks transcriptions used for training. It contains about 2.7M words.

For each of the three sources listed above, we trained a 4-gram backoff LM using the modified shift beta smoothing method as supplied by the IRSTLM toolkit [10]. The three LMs CONTAIN, respectively, about:

- **giga5** 130M bigrams, 231M 3-grams, 422M 4-grams;
- **wmt13** 64M bigrams, 69M 3-grams, 92M 4-grams;
- **ted13** 687K bigrams, 223K 3-grams, 132K 4-grams.

Word pronunciations in the lexicon are based on a set of 45 phones. They were generated by merging different source lexica for American English (LIMSI '93, CMU dictionary, Pronlex). In addition, phonetic transcriptions for a number of missing words were generated by using the phonetic transcription module of the Festival speech synthesis system.

The **wmt13** LM was used to compile a static Finite State Network (FSN) which includes LM probabilities and lexicon for the first two decoding passes. The latter LM was pruned in order to obtain a network of manageable size, resulting in a recognition vocabulary of 200K words and into about: 42M bigrams, 34M 3-grams and 31M 4-grams.

As seen in section 1 the ASR hypotheses generated in the second decoding step were used to automatically select documents from all of the available out-of-domain data, i.e. **giga5** and **wmt13**. To do this we employed a similarity measure based on the well known TFxIDF (term frequencies x inverse document frequencies) [11] features. More specifically, we selected 100M of words for each given TED talk and trained a corresponding talk-dependent LM (in the following we will refer the latter with **aux100M**). Details of the automatic selection approach can be found in [12].

3.4. Word graphs rescoring

Word graphs are generated in the second decoding step. To do this, all of the word hypotheses that survive inside the trellis during the Viterbi beam search are saved in a word lattice containing the following information: initial word state in the trellis, final word state in the trellis, related time instants and word log-likelihood. From this data structure and given the LM used in the recognition steps, WGs are built with separate acoustic likelihood and LM probabilities associated to word transitions. To increase the recombination of paths inside the trellis and consequently the densities of the WGs, the so called word pair approximation [13] is applied. In this way the resulting graph error rate was estimated to be 6.0% on the development set used for IWSLT2011 evaluation campaign (i.e. 19 TED talks), about $\frac{1}{3}$ of the corresponding WER, that resulted to be 17.6%.

WGs are rescored using an interpolated LM that combine all of the four LMs described above, **giga5**, **wmt13**, **aux100M** and the in-domain LM **ted13**. To do this, the original LM probability on each arc of each WG is substituted with the linearly interpolated probability. Note that the development set used to train the interpolation weights is again the ASR output of the second decoding step and, therefore, talk specific interpolation weights are estimated. Note also that acoustic model probabilities associated to arcs of WGs remain unchanged.

In addition WGs were rescored using singularly each one of the above mentioned LMs, thus obtaining 5 different outputs for each automatically transcribed talk (including the ones obtained with the interpolated LM). These latter ASR output hypotheses have been combined, using ROVER, in order to produce the final submission. Note that the latter final ROVER combination makes use of word confidence measures.

4. German transcription system

German ASR makes only use of first and second decoding passes described for English ASR. For German we didn't perform any data selection, in order to build focused LMs, as well as any WG rescoring step.

4.1. AM training

German AMs were trained using Euronews videos downloaded in the last few years from the portal http://de.euronews.com/. Each video has associated a reference text, that could be just a summary, an accurate transcription of the news, or the transcription of a part of the news. We apply lightly supervised training, in a way similar to that described for English ASR, to select segments for training. Three iterations have been used before stopping the selection process, resulting into about 256 hours of training audio data, including silences.

4.2. Linguistic processing and LM training

In German, compound words are a significant percentage of the common lexicon, and should be taken into account to avoid unacceptable out-of-vocabulary (OOV) rate. We built an automatic system that, given a lexicon of German words ordered by frequency, decides which words have to be considered as compounds and propose a splitting.

We extracted form the lexicon a set of words that can

be considered "basewords". These latter words are shorter than a predefined threshold (e.g. 15 characters) and exhibit a frequency higher than another threshold (e.g. greater than 2).

Then we defined, by hand, a "falsebasewords" file which contain some acronyms (17 in the actual version, namely: der die das er es sc sch fts ic des wal sge ger cht ati rwe ler) than cannot be basewords but that were frequently observed. The defined acronyms are used to form wrong decompositions.

Finally, an algorithm was implemented that detects if a suspected compound word can be obtained by concatenating basewords. Among the possible decompositions, the one is chosen which minimizes a cost function favoring longer words. Some German compound rules were added to the algorithm, that basically allow the insertion of the suffixes "s","n","es","en". A sample of decompositions is given in Table 1.

compound word	decomposition
Krankenversicherung	kranken+ +Versicherung
Ministerpräsidenten	Minister+ +Präsidenten
Bundesgeschäftsführer	Bundes+ +Geschäfts+ +Führer
Sicherheitskonferenz	Sicherheits+ +Konferenz
Auseinandersetzungen	auseinander+ +Setzungen
Bundesverfassungsgericht	Bundes+ +Verfassungs+ +Gericht
Oberstaatsanwaltschaft	Oberstaatsanwaltschaft

Table 1: Example of compound word decomposition.

Finally, a method was implemented to join compound words after ASR.

A German 4-gram LM was trained after the split of numbers and compound words on a corpus, formed by crawled news and European Parliament transcriptions, containing about 1.6G of words. Cut-off date was end of June 2012. In-domain text data have also been used for LM adaptation.

5. Official results

Final results (%WER), after adjudication, of the English system for:
tst2011, primary 13.6%
tst2012, primary 16.2%
tst2013, primary 23.2%.

Final result, after adjudication, of the German system for:
tst2013, primary 37.5%.

6. Conclusions

We presented descriptions of our ASR systems used to submit runs to the IWSLT2013 Evaluation Campaign for both English and German audio track. Both systems were trained applying lightly supervised training to audio data that do not have associated "accurate" transcriptions.

English ASR system makes also use of a procedure that allows to rescore WGs with a combination of several LMs, some of them trained on sets of automatically selected data.

7. Acknowledgements

This work was partially supported by the European project EU-BRIDGE, under the contract FP7-287658.

8. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc. of the International Workshop on Spoken Language Translation*, Hong Kong, HK, December 2012.

[2] L. Lamel, J. Gauvain, and G. Adda, "Investigating lightly supervised acoustic model training," in *Proc. of ICASSP*, vol. 1, Salt Lake City, USA, 2001, pp. 477–480.

[3] D. Falavigna, G. Gretter, F. Brugnara, and D. Giuliani, "Fbk @ iwslt 2012 - asr track," in *Proc. of the International Workshop on Spoken Language Translation*, Hong Kong, HK, December 2012.

[4] J. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recogniser Output Voting Error Reduction (ROVER)," in *Proc. of ASRU*, Santa Barbara, CA, 1997, pp. 347–352.

[5] M. Cettolo, "Segmentation, classification and clustering of an italian broadcast news corpus," in *Proc. of Content-Based Multimedia Inf. Access Conf. (RIAO)*, Paris, France, 2000, pp. 372–381.

[6] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[7] G. Stemmer, F. Brugnara, and D. Giuliani, "Using Simple Target Models for Adaptive Training," in *Proc. of ICASSP*, vol. 1, Philadelphia, PA, March 2005, pp. 997–1000.

[8] D. Giuliani, M. Gerosa, and F. Brugnara, "Improved automatic speech recognition through speaker normalization." *Computer Speech and Language*, vol. 20, no. 1, pp. 107–123, Jan. 2006.

[9] D. Giuliani and F. Brugnara, "Experiments on Cross-System Acoustic Model Adapatation," in *ASRU Workshop 2007*, Kyoto, Japan, Dec. 2007, pp. 117–122.

[10] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models," in *Proc. of INTERSPEECH*, Brisbane, Australia, September 2008, pp. 1618–1621.

[11] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," in *First International Conference on Machine Learning*, New Brunswick:NJ, USA, 2003.

[12] D. Falavigna and G. Gretter, "Focusing language models for automatic speech recognition," in *Proc. of the International Workshop on Spoken Language Translation*, Hong Kong, HK, December 2012.

[13] X. Aubert and H. Ney, "A word graph algorithm for large vocabulary continuous speech recognition," in *Proc. of ICSLP*, 1994, pp. 1355–1358.

QCRI at IWSLT 2013: Experiments in Arabic-English and English-Arabic Spoken Language Translation

Hassan Sajjad, Francisco Guzmán, Preslav Nakov, Ahmed Abdelali, Kenton Murray, Fahad Al Obaidli, Stephan Vogel

Qatar Computing Research Institute
Qatar Foundation

{hsajjad,fguzman,pnakov,aabdelali,kwmurray,faalobaidly,svogel}@qf.org.qa

Abstract

We describe the Arabic-English and English-Arabic statistical machine translation systems developed by the Qatar Computing Research Institute for the IWSLT'2013 evaluation campaign on spoken language translation. We used one phrase-based and two hierarchical decoders, exploring various settings thereof. We further experimented with three domain adaptation methods, and with various Arabic word segmentation schemes. Combining the output of several systems yielded a gain of up to 3.4 BLEU points over the baseline. Here we also describe a specialized normalization scheme for evaluating Arabic output, which was adopted for the IWSLT'2013 evaluation campaign.

1. Introduction

We describe the Arabic-English and English-Arabic statistical machine translation (SMT) systems developed by the Qatar Computing Research Institute (QCRI) for the 2013 open evaluation campaign on spoken language translation organized in conjunction with the International Workshop on Spoken Language Translation (IWSLT). Below we give an overview of the settings we experimented with:

- **Decoders:** We used a phrase-based SMT (PBSMT), as implemented in Moses [1], and two hierarchical decoders: Jane [2] and cdec [3]. See Section 6 for details.
- **Decoder settings:** There are a variety of settings available for the above decoders. We explored a number of them, most notably, operation sequence model, minimum Bayes risk decoding, monotone-at-punctuation, dropping out-of-vocabulary words, etc. We selected to retain those settings that improved the overall translation quality as measured on the dev-test set. See Section 4 for further details.
- **Arabic segmentation:** To reduce data sparseness, Arabic words are typically segmented into multiple tokens, e.g., by segmenting out conjunctions, pronouns, articles, etc. We experimented with standard segmentation schemes such as D0, D1, D2, D3, S2 and ATB, as defined in MADA [4, 5]. See Section 5 for details.
- **Domain adaptation:** We experimented with three domain adaptation methods to make better use of the huge UN data, which is out-of-domain: (i) Modified Moore-Lewis filtering, (ii) phrase table merging, and (iii) phrase table backoff. See Section 7 for details.

For our final submission, we synthesized a translation by combining the output of our best individual system with the output of other systems that are both relatively strong and can contribute to having more diversity, e.g., using a different decoder or a different segmentation scheme.

We achieved the most notable improvements in terms of BLEU when translating from Arabic-to-English using an operation sequence model (+0.6 BLEU on tst2010), phrase table merging and phrase table backoff (+0.6 BLEU), interpolated language model (+1.5 BLEU), and system combination using different decoders and different segmentation schemes (+0.6 BLEU). For the English-to-Arabic direction, we observed smaller improvements compared to the reverse direction, but there the absolute baseline was also much lower.

Finally, we proposed normalization for Arabic output evaluation, which was adopted as official for IWSLT'2013.[1]

2. Data

For the Arabic-English language pair, the IWSLT'2013 training data consisted of a small in-domain bitext, i.e. the TED talks[2] (IWSLT), and a large out-of-domain bitext, i.e. the multiUN corpus (UN). There were also tuning and development bitexts: dev2010 and tst2010. Conversely, for language modeling, a larger number of monolingual corpora were permissible. They are all listed in Table 1, together with their corresponding word count statistics.

3. Baseline

Data. We built a baseline system using the Moses toolkit and the IWSLT training data only, i.e., the TED talks. At development time, we tuned and tested on the provided dev2010 and tst2010 datasets.

[1] The normalizer is freely available at http://alt.qcri.org/tools/.

[2] https://wit3.fbk.eu/mt.php?release=2013-01

Monolingual corpora	# Words
English	
IWSLT mono	2.7M
10^9 English-French	575M
SETimes	4.2M
UN (Es-En + En-Fr)	597M
UN (Ar-En)	115M
News Crawl 2007-2009	643M
News Crawl 2009-2012	745M
Common Crawl	185M
Wiki Headlines	1.1M
Europarl v.7	54M
News Commentary v.8	5.3M
Gigaword v.5	4,032M
Arabic	
IWSLT mono	2.7M
UN	134M
News Commentary Arabic v.8	4.8M
Gigaword Arabic v.5	1,373M

Table 1: Admissible training data for language modeling. Here English is tokenized, and Arabic is ATB-segmented.

Preprocessing. We segmented the Arabic side of the bitext following the ATB scheme and using the Stanford word segmenter [6]. For the English side, we used the standard tokenizer of Moses, and we further applied truecasing/lowercasing when English was the target/source language.

Training. We built separate directed word alignments for English-to-Arabic and for Arabic-to-English using IBM model 4 [7], and we symmetrized them using the *grow-diag-final-and* heuristics [8]. We then extracted phrase pairs with a maximum length of seven, and we scored them using maximum likelihood estimation with Kneser-Ney smoothing, thus obtaining a phrase table where each phrase pair has the standard five translation model features. We also built a lexicalized reordering model [9]: *msd-bidirectional-fe*. For language modeling, we used KenLM [10] to build a 5-gram Kneser-Ney smoothed model, trained on the target side of the training bi-text. Finally, we built a large joint log-linear model, which used standard PBSMT feature functions: language model probability, word penalty, the parameters from the phrase table, and those from the reordering model.

Tuning. We tuned the weights in the log-linear model by optimizing BLEU [11] on the tuning dataset, using PRO [12]. We allowed the optimizer to run for up to 10 iterations, and to extract 1000-best lists on each iteration.

Decoding. On tuning and testing, we used monotone-at-punctuation. On testing, we further used cube pruning.

Table 2 shows the results[3] for the baseline English-to-Arabic and Arabic-to-English SMT systems, compared to the baseline results reported on the WIT[3] webpage.

[3] For tst2010, we report MultEval BLEU and TER0.8: on tokenized and recased output for English, and on QCRI-normalized output for Arabic. For tst2011, tst2012, and tst2013, the organizers used slightly different scorers.

	Arabic-English		English-Arabic	
System	BLEU	1-TER	BLEU	1-TER
IWSLT baseline	23.6	43.0	11.9	28.6
Our baseline	24.7	45.6	12.6	29.1

Table 2: Our vs. IWSLT baseline results for English-to-Arabic and Arabic-to-English SMT, evaluated on tst2010.

4. System Settings

Below we discuss the decoder settings and extensions we experimented with, focusing on Arabic-to-English. Table 3 shows the impact of each feature when added to the baseline.

Tuning. [13] have shown that PRO tends to generate too short translations.[4] They have suggested that the root of the problem was that PRO optimizes sentence-level BLEU+1, which smooths the precision component of BLEU, but leaves the brevity penalty intact, which destroys the balance between them. They have proposed a number of fixes, the simplest and most efficient among them being to smooth the brevity penalty as well.[5] In our experiments, this yielded +0.2 BLEU for Arabic-to-English on tst2010.

Operation sequence model. The operation sequence model (OSM) is an n-gram-based model, which represents the aligned bitext into a sequence of operations, e.g., generate a sequence of source and target words or perform reordering. The model memorizes Markov chains over such sequences, thus fusing lexical generation and reordering into a single generative model. OSM offers two advantages. First, it considers bilingual contextual information that goes beyond phrase boundaries. Second, it provides a better reordering mechanism that has richer conditioning than a lexicalized reordering model: the probability of an operation is conditioned on the n previous translation and reordering decisions. We used the Moses implementation of OSM [15], which has yielded improvements at WMT'13 [16]. In our experiments, it yielded +0.6 BLEU for Arabic-to-English on tst2010.

Minimum Bayes risk decoding. We also experimented with minimum Bayes risk decoding (MBR)[17], which, instead of outputting the translation with the highest probability, prefers the one that is most similar to best n translations. In our case, using MBR did not improve over the baseline.

Translation options per input phrase. By default, Moses uses up to 20 translation options per input phrase, but [16] have shown better results with 100. In our experiments, this yielded +0.1 BLEU for Arabic-to-English on tst2010.

Transliterating OOVs. Out-of-vocabulary (OOV) words are problematic for languages with different scripts. Thus, we tried transliteration as post-processing: we extracted 1-1 word alignments from a subset of the UN bitext, and we used them to train a character-level transliteration system [18, 19] using Moses. As Table 3 shows this did not help, probably due to the small number of OOVs in tst2010.

[4] See [14] for a discussion about more potential issues with PRO.

[5] Available in Moses: `--proargs='--smooth-brevity-penalty'`

System	Arabic-English (tst2010) BLEU	1-TER
Baseline (B)	24.7	45.6
OSM	25.3	46.1
MBR	24.7	45.7
Ttable 100	24.8	45.6
PRO-fix [13]	24.9	44.7
TRANSLIT	24.7	45.6
Drop UNK	24.8	45.7

Table 3: Impact of each feature when added to the baseline.

Dropping OOVs. An alternative to transliteration is to just drop all OOV words as part of the decoding process. We did this on both tuning and testing, and it yielded +0.2 BLEU for Arabic-to-English on tst2010.

Language model. For language modeling (LM), we used most of the available data shown in Table 1, processed with the Moses tokenizer for English, and with the Stanford ATB segmenter for Arabic. For each data source, we trained a separate 5-gram LM with Kneser-Ney smoothing. We then interpolated these models, minimizing the perplexity on the target side of dev2010.[6] Finally, we binarized them using KenLM [10] with probing and no quantization. Table 4 shows that using these LMs yields +1.5 BLEU for English, but only +0.6 for Arabic; this is probably due to less data being available for Arabic LM training.

System	BLEU tst2010 Arabic-English	English-Arabic
Baseline (TED LM)	24.7	10.6
Large LM	26.2	11.2

Table 4: The impact of using a large LM on tst2010.

5. Arabic Segmentation

In Arabic, various clitics such as pronouns, conjunctions and articles appear concatenated to content words such as nouns and verbs. This can cause data sparseness issues, and thus clitics are typically segmented in a preprocessing step. There are various standard segmentation schemes defined in MADA [4, 5] such as D0, D1, D2, D3 and S2, for which we used the MADA+TOKAN toolkit [20], as well as ATB, which we performed using the Stanford segmenter [6]. Table 5 shows the results when training on the TED bitext only. We can see that ATB performed the best overall with a BLEU score of 24.7, followed by S2 with a score of 24.5.

[6]For Gigaword, a preliminary interpolation between models computed over two-year partitions of the corpus (e.g., 2005 and 2006) was necessary because of memory limitations of the machines we used to train the LMs.

System	Arabic-English (tst2010) BLEU	1-TER
SEG-D0	22.4	43.0
SEG-D1	23.6	44.2
SEG-D2	24.1	45.2
SEG-D3	24.4	45.5
SEG-S2	24.5	45.7
SEG-ATB	24.7	45.6

Table 5: Using different Arabic segmentation schemes.

System	Arabic-English (tst2010) BLEU	1-TER
Moses PBSMT	24.7	45.6
cdec	24.3	44.6
Jane	24.1	43.6

Table 6: Baseline results with different decoders.

6. Decoders

In our experiments, we used several decoders. Table 6 shows the baseline results for each of them.

Moses PBSMT. We used the phrase-based model as implemented in Moses [1]. It is described in our baseline above.

cdec. We further experimented with the hierarchical cdec decoder [3]. We used its default features: forward and backward translation features, singleton features, a glue-rule probability, and a pass-through feature (to handle OOVs). We tuned the parameters using MIRA with IBM BLEU as the objective function and a k-best forest size of 250.

Jane. We also used another hierarchical phrase-based decoder: Jane 2.2 [2]. We used the standard features: phrase translation probabilities and lexical smoothing in both directions, word and phrase penalties, a distance-based distortion model, and a 5-gram LM. We optimized the weights using MERT [21] on 100-best candidates with BLEU as objective.

7. Adaptation

The IWSLT dataset contains a small in-domain corpus (TED talks) and a large out-of-domain corpus (UN). In this section, we explore various ways to make best use of the out-of-domain data to improve the baseline system.

7.1. Modified Moore-Lewis Filtering (MML)

Moore and Lewis [22] presented a method for selecting relevant sentences from out-of-domain data for language modeling. Axelrod et al. [23] further extended it to parallel corpora, considering both the source and the target side of the bi-text, as well as in-domain and out-of-domain data, when scoring each sentence pair; their method is known as modified Moore and Lewis, or MML. They have shown that MML can yield improvements in SMT quality when selecting as little as just 1% of the out-of-domain training bi-text.

System	Training	BLEU	1-TER
baseline	IWSLT	24.7	45.6
MML1	IWSLT+2%UN	24.4	45.6
MML2	IWSLT+3%UN	24.4	45.6
MML3	IWSLT+4%UN	24.3	45.1
MML4	IWSLT+5%UN	24.2	45.6
MML5	IWSLT+100%UN	21.9	42.8

Table 7: Arabic-to-English: training on the IWSLT bi-text plus various MML-filtered UN bi-texts.

We experimented with MML, selecting varying percentages of out-of-domain UN data. Note that this additional data impacts all models: the translation model, the reodering model, and the language model. However, in order to allow for more fair head-to-head comparison, in Table 7 we show experimental results where we limit the LM training data to IWSLT only. We can see that each MML-adapted system suffers a drop in performance compared to the baseline system, which can be attributed to the differences between the in-domain and the out-of-domain data in terms of sentence structure, vocabulary, and style. Note that using just 2% and 3% of UN data works best, but this is still worse than not using UN data at all.

7.2. Merging Translation and Reordering Models

Given the negative results with MML, we also tried an alternative way to make use of the out-of-domain UN data, namely phrase table merging as described in [24, 25]. In the merged phrase table, we kept either (a) both phrases, or (b) the one coming from the in-domain data only. In either case, we added three additional binary features for each phrase pair indicating whether it came from (i) the in-domain data, (ii) the out-of-domain data, and (iii) both. Similarly, we merged reordering models, where we preferred the scores from the in-domain model. We further experimented with merging a phrase table for IWSLT with one for 3% of UN.

The results are shown in Table 8; note that this time we use the large interpolated language model presented in Table 4.. We show results for merging IWSLT with 3% of the UN data (MER1, MER2) as well as with the full UN (MER3, MER4), with duplicates kept (MER1, MER3) or removed (MER2, MER4). For comparison, we also show the baseline of using IWSLT only. We can see that using the full UN data works best, yielding +0.6 BLEU points of improvement.

7.3. Backoff Phrase Tables

The Moses toolkit allows for the use of a *backoff* phrase table in addition to a *main* phrase table. The phrases from the backoff phrase table are used when the translation of a phrase is unknown to the main phrase table. The backoff order determines the maximum phrase length for which this operation is allowed.

System	Training	BLEU	1-TER
baseline	IWSLT	26.2	46.6
MER1	IWSLT & 3%UN	26.2	46.4
MER2	IWSLT & 3%UN, no-dup	26.5	46.7
MER3	IWSLT & UN	26.6	47.0
MER4	IWSLT & UN, no-dup	26.8	47.1

Table 8: Arabic-to-English: phrase table merging.

In our experiments, we considered the phrase table built using the in-domain data as the main phrase table, and that built using the full UN data as the backoff phrase table. We tried n-grams of different orders for the backoff. Table 9 shows the results for backoff orders of 4, 5 and 6; again, we use the large interpolated language model presented in Table 4.. We can see that backoff orders of 4 and 5 performed best, achieving results that are very similar to what we obtained with phrase tables merging: comparing Table 9 to Table 8, we see the same BLEU score of 26.8, and a bit different 1-TER score. We believe that this indicates that the UN data is mostly useful for specific cases, e.g., to translate unknowns, but that it should not be blindly concatenated to the in-domain data because this hurts the performance.

System	Backoff order	BLEU	1-TER
baseline	0	26.2	46.6
BO1	4	26.8	47.2
BO2	5	26.8	47.2
BO3	6	26.7	47.2

Table 9: Arabic-to-English: phrase table backoff.

7.4. Best Adaptation

In the remainder of this paper, we will consider the MER4 system as our best adapted system. Note that when we also use OSM trained on the IWSLT bi-text, the BLEU score further improves by +0.6 points. Table 10 shows these results.

System	BLEU	1-TER
MER4	26.8	47.1
MER4+OSM$_{in}$	27.4	47.9

Table 10: Arabic-to-English: our best adapted system MER4 combined with OSM.

8. Arabic-to-English Machine Translation

We built several Arabic-to-English SMT systems based on the settings described in the previous sections; we further used system combination to produce our final translation. Below we give details about the individual systems.

System	Training	BLEU	1-TER
SEG-D1	IWSLT-3%UN	25.5	45.7
SEG-D2	IWSLT-3%UN	26.3	46.5
SEG-D3	IWSLT-3%UN	26.4	47.2
SEG-S2	IWSLT-3%UN	26.7	47.3
SEG-ATB	IWSLT-3%UN	27.0	47.4
cdec	IWSLT	25.4	45.4
cdec-UN	IWSLT-3%UN	25.3	45.6
Jane	IWSLT	24.7	42.5
FF	**IWSLT-100%UN**	**27.5**	**47.9**

Table 11: Arabic-to-English SMT systems (tst2010).

Segmentation. We built five phrase-based SMT systems, each using a different MADA segmentation scheme for the Arabic side: D1, D2, D3, S2 and ATB. We did not segment the complete UN data with each of these segmentation schemes due to time constraints. Instead, we used the 3% UN data filtered using MML to build a phrase table, which we then merged with the phrase table for IWSLT, preferring IWSLT phrase pairs in case of duplicates; this yielded systems coresponding to the MER2 line in Table 8. We further used OSM and MBR.

Decoder. We used three decoders: one phrase-based (Moses) and two hierarchical (cdec and Jane). Note that most of the settings described in Section 4 are applicable to the phrase-based decoder only. We trained cdec and Jane on the IWSLT data only, while still using the large interpolated LM. For cdec, we further built another system which was trained on a concatenation of the IWSLT data and the 3% UN data.

Full featured run. Finally, we further extended the MER4-OSM$_{in}$ system (see Table 10), which uses the complete UN data and the adapted OSM, with two additional settings: (i) MBR and (ii) ttable 100. This is our best indiviual run that does not use system combination, which we will call Full Featured (FF) below. We submitted it as our contrastive run to the competition.

Table 11 summarizes the results for all our Arabic-to-English SMT systems.

8.1. System Combination Results

We recombined hypotheses produced by various subsets of the systems in Table 11 using the Multi-Engine MT system (MEMT) [26]. The results are presented in Table 12. We can see that combining all segmentations yields +0.4 BLEU over our best individual system FF. Further adding cdec to the combination, yields another +0.2 BLEU; this was our primary system for Arabic-to-English.

8.2. Official Results

Table 13 shows the official results of our Arabic-to-English contrastive and primary systems. PRM is our primary system, a system combination of all systems in Table 11.

System	BLEU	1-TER
FF	27.5	47.9
FF, SEG-ALL	27.9	47.4
FF, cdec-UN	27.7	47.2
FF, cdec-UN, Jane	27.6	47.4
FF, SEG-ALL, cdec, cdec-UN	**28.1**	**47.6**

Table 12: Arabic-to-English syscomb (tst2010).

System	tst2011		tst2012		tst2013	
	BLEU	1-TER	BLEU	1-TER	BLEU	1-TER
FF	26.9	44.8	28.7	49.7	30.0	48.9
PRM	27.8	44.8	30.3	50.5	30.5	48.6

Table 13: Arabic-to-English: official scores (mteval-v13a).

9. English-to-Arabic Machine Translation

For English-to-Arabic translation, we experimented with different segmentation schemes: D0, D1, D2, D3, S2 (using MADA), and ATB (using the Stanford segmenter). Note that this is more complicated here than for Arabic-to-English because the segmentation is on the target side; thus, for English-to-Arabic SMT, there is need for (i) a separate LM for each segmentation, and (ii) desegmentation of the output.

A separate LM for each segmentation. Since the segmentation is on the target side, it applies to the language model as well. This means that if we wanted to experiment with different segmentations, we needed a separate language model for each of them, which is time- and resource-consuming. In practical terms, this prevented us from building strong language models for D0, D1, D2, D3 and S2, for which we used an LM trained on the Arabic side of the IWSLT bi-text only. It was for the ATB segmentation only that we could build a strong LM through interpolation, similarly to our Arabic-to-English LM, that also used the Gigaword Arabic, UN, and News Commentary data (see Table 1).

Desegmentation. Unlike the Arabic-to-English direction, where the segmentation was on the input side and thus the output was unaffected, here the segmentation had to be undone. For example, if we use an ATB-segmented target side, we end up with an ATB-segmented translation output, which we have to desegment in order to obtain proper Arabic. Desegmentation is not a trivial task since it involves some morphological adjustments, see [27] for a broader discussion. For desegmentation, we used the best approach described in [27]; in fact, we used their implementation.

Normalization. Translating into Arabic is tricky because the Arabic spelling is often inconsistent in terms of punctuation (using both Arabic UTF8 and English punctuation symbols), digits (appearing as both Arabic and Indian characters), diacritics (can be used or omitted, and can often be wrong), spelling (there are many errors in the spelling of some Arabic characters, esp. *Alef* and *Ta Marbuta*; also, *Waa*

appears sometimes separated). These problems are especially frequent in informal texts such as TED talks. Thus, we normalized Arabic to make it more consistent. We first concatenated back the conjunction *Waa* when detached (it is almost never detached in proper Arabic). We then used MADA to normalize the following: (i) punctuation: converted Arabic UTF8 punctuation to English, (ii) digits: converted all Indian digits to the standard Arabic digits 0,1,. . .,9, (iii) diacritics: dropped them all, (iv) spelling: fixed potential errors in the different forms of *Alef*, *Alef Maqsura*, *Ta Marbuta*, etc. Finally, we converted all instances of "..", which are common in informal Arabic text, but are never used in English, to "...".

Tokenization and detokenization. We further had to perform tokenization and detokenization. Note that this is different from segmentation: segmentation is about splitting words into multiple words, while tokenization is mainly about separating punctuation from words. For tokenization, we used the Europarl tokenizer: note that it does not work on general Arabic text (e.g., because it cannot handle the UTF8 Arabic punctuation symbols), but it works just fine on our normalized Arabic. For detokenizing the final Arabic desegmented output, we used the Moses detokenizer; again, it only works because it sees normal English punctuation.

Scoring the Arabic SMT output. While the systems participating in IWSLT'2013 were supposed to output proper Arabic, directly scoring their output against the references with the NIST scoring tool v13a is problematic because of the above-described inconsistencies in Arabic, which also happen in the references for the tuning and the testing sets (in addition to training). Since these variations are quite random and depend on the style of the author of each piece of text, it does not make sense for a translation system to try to model them. Yet, they can affect evaluation scores a lot![7] Thus, we normalize both the SMT output and the reference with the QCRI normalizer: it applies the above-described normalization and also performs tokenization. Then, we calculate a BLEU and a TER score using MultEval, which does not perform internal tokenization (unlike the NIST scoring tool). This scoring procedure is official for the English-to-Arabic translation direction at IWSLT'2013.

9.1. Individual and Combined Systems

The results for the individual systems are shown in Table 14. We can see that ATB performs best, which is to be expected since it uses a much larger LM. However, adding the UN bitext in phrase table combination had a very minor impact on BLEU, only adding +0.2 points to FF.

Similarly to the Arabic-to-English system, we used MEMT to combine the outputs of several systems. The challenge was to make these outputs compatible: they were to be (1) desegmented, and (2) re-segmented using the ATB scheme. This allowed us to perform system combination using the large Arabic ATB language model.

[7]E.g., the score for the organizer's baseline system goes up from 9.61 (after tokenization with Europarl) to 11.89 when using the QCRI normalizer.

System	Training	BLEU	1-TER
SEG-D0	IWSLT	12.3	30.2
SEG-D1	IWSLT	12.6	30.6
SEG-D2	IWSLT	12.5	30.7
SEG-D3	IWSLT	12.5	30.5
SEG-S2	IWSLT	12.5	30.2
SEG-ATB	IWSLT, big-LM	13.6	31.3
cdec	IWSLT	12.7	29.8
Jane	IWSLT	12.2	28.8
FF	**IWSLT+UN, big-LM**	**13.8**	**31.4**

Table 14: English-to-Arabic SMT systems (tst2010).

System	BLEU	1-TER
FF	13.8	31.4
FF, SEG-ALL, cdec	**13.7**	**30.2**

Table 15: English-to-Arabic syscomb (tst2010).

We tried many system combinations, but we were unable to improve over FF. Table 15 shows our best combination; even though it yielded -0.1 BLEU points on tst2010, we submitted it as primary, to be consistent with Arabic-to-English.

9.2. Official Results

Table 16 shows the official results of our English-to-Arabic contrastive and primary runs. We can see that the system combination performed slightly better, after all.

10. English-to-Arabic Spoken Translation

Translating the ASR output poses several additional challenges over translating properly transcribed text such as (1) finding sentence boundaries, (2) restoring case, and (3) restoring punctuation. Note that for this year's competition, speech segmentation was provided by the organizers, which solves (1). We further trained our English-to-Arabic SMT system on lowercase English input, thus eliminating the need for (2). Lastly, we addressed (3) by considering two levels of punctuation restoration. As a baseline, we just inserted a full stop at the end of each sentence. Next, we treated punctuation marks as hidden events occurring between words. Thus, the problem was reduced to finding the most likely tag sequence using an n-gram language model.

System	tst2011		tst2012		tst2013	
	BLEU	1-TER	BLEU	1-TER	BLEU	1-TER
FF	15.15	31.66	15.68	35.28	15.68	35.82
PRM	15.54	30.81	15.54	34.43	15.78	34.57

Table 16: English-to-Arabic: our official results (calculated using the QCRI normalizer, then MultEval).

For this purpose, we used the *hidden-ngram* tool from the SRILM toolkit [28]. We trained the LM on the tokenized monolingual English portion of the IWSLT training data. The list of punctuation marks (tags) included the following: *comma (,), semi-colon (;), colon (:), quotation marks (”), question marks (?), period (.),* and *ellipsis (...)* .

For our contrastive SLT system, we reused the best English-to-Arabic system from the previous section (FF). Table 17 shows the results for different methods for punctuation restoration. Note that decoding with a simple full stop addition improved the score by about +1.3 BLEU points. Further restoring the rest of the punctuation marks yielded an additional improvement of +1.3 BLEU points. As a reference, we also include the *Oracle* input, i.e., the MT text input (with the same sentence segmentation as the ASR's 1-best).

System	**tst2010**	
	BLEU	1-TER
Raw 1-best input	6.2	21.1
+ full stop at the end	7.5	23.6
+ punctuation restoration	8.8	23.7
Text input (Oracle)	14.0	31.3

Table 17: English-to-Arabic SLT: punctuation restoration.

10.1. System Combination Results

Similarly to the English-to-Arabic text translation, we used MEMT to combine the output of several systems. The combined output yielded +0.1 BLEU points over the best system.

10.2. Official Results

Table 18 shows the official results for our English-to-Arabic SLT submissions: contrastive (FF single-best) and primary (PRM, system combination). The systems are the same as for English-to-Arabic text translation.

System	**tst2013**	
	BLEU	1-TER
FF	10.27	26.24
PRM	10.33	26.28

Table 18: English-to-Arabic SLT: our official results (calculated using the QCRI normalizer, then MultEval).

11. Conclusion

We have presented the Arabic-English and English-Arabic SMT systems developed by the Qatar Computing Research Institute for the IWSLT'2013 evaluation campaign on spoken language translation. We experimented with three decoders and various settings thereof, we tried different domain adaptation methods, and we performed system combination. For the Arabic side, we also used various segmentation schemes.

For domain adaptation, we achieved best results with the full UN data and phrase table merging. The SMT systems built using different MADA segmentation schemes for Arabic (the ATB segmentation was strongest) and using different decoders (Moses performed better than cdec and Jane.) added diversity and were useful for system combination.

For English-to-Arabic, we observed that the gains from the various decoding settings, domain adaptation and system combination were all lower compared to those for the Arabic-to-English system. We plan to investigate this in future work.

Finally, we proposed normalization for Arabic output evaluation, which was adopted as official for IWSLT'2013.

Acknowledgements. We would like to thank Nizar Habash for sharing the Arabic desegmentation code.

12. References

[1] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the Association for Computational Linguistics (ACL'07)*, Prague, Czech Republic, 2007.

[2] D. Vilar, D. Stein, M. Huck, and H. Ney, "Jane: Open source hierarchical translation, extended with reordering and lexicon models," in *Proceedings of the Workshop on Statistical Machine Translation and Metrics MATR (WMT'10)*, Uppsala, Sweden, 2010.

[3] C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, P. Blunsom, H. Setiawan, V. Eidelman, and P. Resnik, "cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models," in *Proceedings of the Association for Computational Linguistics (ACL'10)*, Uppsala, Sweden, 2010.

[4] N. Habash and F. Sadat, "Arabic preprocessing schemes for statistical machine translation," in *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'06)*, New York, NY, USA, 2006.

[5] I. Badr, R. Zbib, and J. R. Glass, "Segmentation for English-to-Arabic statistical machine translation," in *Proceedings of the Association for Computational Linguistics (ACL'08)*, Columbus, OH, USA, 2008.

[6] S. Green and J. DeNero, "A class-based agreement model for generating accurately inflected translations," in *Proceedings of the Association for Computational Linguistics (ACL'12)*, Jeju Island, Korea, 2012.

[7] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[8] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL'03)*, Edmonton, Canada, 2003.

[9] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh system description for the 2005 IWSLT speech translation evaluation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT'05)*, Pittsburgh, PA, USA, 2005.

[10] K. Heafield, "KenLM: Faster and smaller language model queries," in *Proceedings of the Workshop on Statistical Machine Translation (WMT'11)*, Edinburgh, UK, 2011.

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the Association for Computational Linguistics (ACL'02)*, Philadelphia, PA, USA, 2002.

[12] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*, Edinburgh, UK, 2011.

[13] P. Nakov, F. Guzmán, and S. Vogel, "Optimizing for sentence-level BLEU+1 yields short translations," in *Proceedings of the International Conference on Computational Linguistics (COLING'12)*, Mumbai, India, 2012.

[14] P. Nakov, F. Guzman, and S. Vogel, "A tale about PRO and monsters," in *Proceedings of the Association for Computational Linguistics (ACL'13)*, Sofia, Bulgaria, 2013.

[15] N. Durrani, A. Fraser, H. Schmid, H. Hoang, and P. Koehn, "Can Markov models over minimal translation units help phrase-based SMT?" in *Proceedings of the Association for Computational Linguistics (ACL'13)*, Sofia, Bulgaria, 2013.

[16] N. Durrani, B. Haddow, K. Heafield, and P. Koehn, "Edinburgh's machine translation systems for European language pairs," in *Proceedings of the Workshop on Statistical Machine Translation (WMT'13)*, Sofia, Bulgaria, 2013.

[17] S. Kumar and W. Byrne, "Minimum Bayes-risk decoding for statistical machine translation," in *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'04)*, Boston, MA, USA, 2004.

[18] H. Sajjad, A. Fraser, and H. Schmid, "An algorithm for unsupervised transliteration mining with an application to word alignment," in *Proceedings of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'11)*, Portland, OR, USA, 2011.

[19] ——, "A statistical model for unsupervised and semi-supervised transliteration mining," in *Proceedings of the Association for Computational Linguistics (ACL'12)*, Jeju, Korea, 2012.

[20] O. Rambow, N. Habash, and R. Roth, "MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization," in *Proceedings of the International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, 2009.

[21] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the Association for Computational Linguistics (ACL'03)*, Sapporo, Japan, 2003.

[22] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the Association for Computational Linguistics (ACL'10)*, Uppsala, Sweden, 2010.

[23] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*, Edinburgh, UK, 2011.

[24] P. Nakov, "Improving English-Spanish statistical machine translation: experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing," in *Proceedings of the Workshop on Statistical Machine Translation (WMT'08)*, Columbus, OH, USA, 2008.

[25] P. Nakov and H. T. Ng, "Improved statistical machine translation for resource-poor languages using related resource-rich languages," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'09)*, Singapore, 2009.

[26] K. Heafield, G. Hanneman, and A. Lavie, "Machine translation system combination with flexible word ordering," in *Proceedings of the Workshop on Statistical Machine Translation (WMT'09)*, Athens, Greece, 2009.

[27] A. El Kholy and N. Habash, "Orthographic and morphological processing for English–Arabic statistical machine translation," *Machine Translation*, vol. 26, no. 1-2, 2012.

[28] A. Stolcke *et al.*, "SRILM – an extensible language modeling toolkit." in *Proceedings of the International Speech Communication Association (INTERSPEECH'02)*, Denver, CO, USA, 2002.

A Discriminative Reordering Parser for IWSLT 2013

Hwidong Na and Jong-Hyeok Lee

Department of Computer Science and Engineering
Pohang University of Science and Technology (POSTECH), Republic of Korea
{leona, jhlee}@postech.ac.kr

Abstract

We participated in the IWSLT 2013 Evaluation Campaign for the MT track for two official directions: German↔English. Our system consisted of a reordering module and a statistical machine translation (SMT) module under a pre-ordering SMT framework. We trained the reordering module using three scalable methods in order to utilize training instances as many as possible. The translation quality of our primary submissions were comparable to that of a hierarchical phrase-based SMT, which usually requires a longer time to decode.

1. Introduction

Word reordering is one of the most difficult problems in machine translation. Formally, word reordering refers to arrange the source words into a target-like order, i.e. finding a permutation of the source words. Because searching for all possible permutations is an NP-complete problem, statistical machine translation (SMT) systems have restricted their search space for efficiency. For example, the simple distortion model in phrase-based SMT (PBSMT) prohibit a long distance jump beyond a window size during translation. Therefore, PBSMT suffers from the lack of ability for word reordering at a long distance.

Pre-ordering is one of the most prevailing approaches to overcome this limitation of PBSMT. It is a pre-processing method that reorders the source sentence in advance to the later translation using PBSMT. We categorize previous works into three categories. First, pre-ordering using local information reorders either a (flat) word or chunk sequence [1, 2, 3, 4]. Second, pre-ordering using syntactic information manipulates a syntactic tree so that yield a reordered sentence [5, 6, 7, 8, 9]. Third, pre-ordering using an ad-hoc structure for word reordering induces a discriminative parser trained from a parallel corpus, and apply the parser to obtain a reordered source sentence [10, 11].

Both the second and third approaches work with hierarchical structures of the source sentence. While the second approach requires a syntactic parser which might not be available for resource-poor languages, the third one requires only a small manual word aligned corpus in addition to a large parallel corpus. Hereinafter, therefore, we focus on the third approach. Because of the efficiency, the hierarchical structures of the third approach restrict word reordering within a continuous sequence under a sub-structure, i.e., the hierarchical structures obey Inversion Transduction Grammar (ITG) [12] constraints.

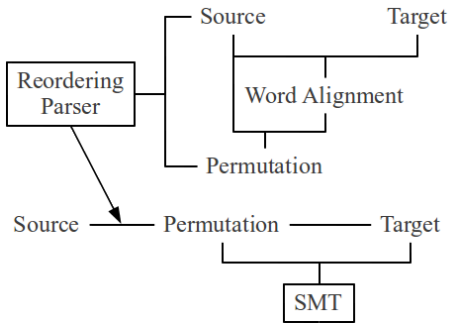


Figure 1: The overall architecture of our system, consisting of a reordering module (reordering parser) and a SMT module

We participated in the IWSLT 2013 Evaluation Campaign for the MT track, and submitted runs for two official directions: German↔English. As German and English have different word orders, we applied a pre-ordering method to resolve the difference requires word reordering.

2. System description

Our system consists of two modules: a reordering module and a SMT module. The reordering module rearranges the words in the source sentence, and the SMT module translates the reordered sentence into the target sentence. The overall architecture of our system is shown in Figure 1. Because we utilized an off-the-shelf SMT system [13] as the SMT module, we focus on the reordering module here.

2.1. Discriminative reordering parser

We briefly summarize the discriminative reordering parser in this section. The most relevant work of this paper was proposed to induce a tree for word reordering produced by a discriminative parser [11]. The goal of their method is to find the best permutation $\hat{\pi}$ for a given source sentence F, according to the following discriminative model.

Algorithm 1: Online learning for a training instance

```
1  procedure UpdateWeight(F, A, w)
2      D ← Parse(F, w)
3      Ḋ ← arg max_{D∈D} Score(D|F, w) + L(D|F, A)
4      D̂ ← arg min_{D∈D} L(D|F, A) − αScore(D|F, w)
5      if L(D̂|F, A) ≠ L(Ḋ|F, A) then
6          w ← β(w + γ(φ(D̂, F) − φ(Ḋ, F)))
7      end
```

$$\begin{aligned} \hat{\pi} &= \arg\max_{\pi} Score(\pi|F) \\ Score(\pi|F) &= Score(D|F, \mathbf{w}) \\ &= \sum_i w_i \phi_i(D, F) \end{aligned} \quad (1)$$

where D is a reordering tree for word reordering which yields π, and w_i and ϕ_i are the ith feature weight and function, respectively.

To learn the weight vector $\mathbf{w}$, they used the loss-driven large-margin training [14] by finding $\dot{D}$ with the highest model score (Eq. 1) and $\hat{D}$ with the smallest loss. A loss function $L(D|F, A)$ is defined by word alignment A, where [11] suggested two kinds of loss functions. Finally, the weight is updated using the difference between the model parse $\dot{D}$ and the oracle parse $\hat{D}$. It is computationally intractable to search over all possible permutations for π. Hence, $\dot{D}$ and $\hat{D}$ are selected among K-best parses encoded in a hyper graph $\mathcal{D}$. To break the tie, $Score(D|F, \mathbf{w})$ and $L(D|F, A)$ are mutually augmented when selecting $\dot{D}$ and $\hat{D}$. The online learning for a training instance $\langle F, A \rangle$ with the current weight vector $\mathbf{w}$ is shown in Algorithm 1 (taken from [11]).

2.2. Scalable training method

We illustrate three methods to scale up the online learning method which iterates several epochs over the training instances. First, we adopted a faster search algorithm known as Cube Growing, and integrated it into a parallel CYK parsing method. Second, the feature generation process run in parallel because it is a major bottleneck of parsing efficiency. Third, the features generated at the first iteration are stored on disk and used in the remaining epochs. In a consequence, our proposed methods enable us to utilize tens of thousands training instances in our experiments.

2.2.1. Cube Growing in parallel CYK parsing

Cube Growing is a dynamic programming algorithm for searching over a hyper graph, proposed by [15]. It produces the kth-best parse on-the-fly, and thus does not enumerate unnecessary hypotheses during the search process. More specifically, two data structures manage the hypothe-

Algorithm 2: Cube Growing in parallel CYK parsing

```
1  procedure Parse
   input : A sentence w_1 ... w_N
   output: A hyper graph with K-best parses
2    for L ∈ [1, N] do
3      for l ∈ [0, N − L]        // in parallel
4      do
5        r ← l + L
6        ModifiedCubeGrowing(l, r)
7      end
8      wait for terminating the cell-level parallelization
9    end
10   root ← The root cell
11   for k ∈ [1, K] do
12     LazyKthBest(root.q, k)
13   end
14
15 procedure ModifiedCubeGrowing
   input : A cell covering [l, r]
   output: A priority queue q with candidates
16   for m ∈ (l, r] do
17     left ← cell [l,m]
18     right ← cell [m,r]
19     L ← peek(left.q)
20     R ← peek(right.q)
21     push(q, Hyp(L, R))        // straight
22     push(q, Hyp(R, L))        // inverted
23   end
24
25 procedure LazyKthBest
   input : A priority queue q and the demanded k
   output: The kth hypothesis in b
26   b ← the list of best hypothesis
27   while size(b) < k + 1 and size(q) > 0 do
28     best ← pop(q)
29     LazyNext(q, best)
30     push(b, best)
31   end
32
33 procedure LazyNext
   input : A priority queue q and the hypothesis best
   output: An extended priority queue q'
     /* best.L and best.R are the left
        and right children of best,
        respectively                    */
34   L ← LazyKthBest(left, rank(best.L)+1)
35   if L exists then
36     push(q, Hyp(L, best.R))   // straight
37     push(q, Hyp(best.R, L))   // inverted
38   end
39   R ← LazyKthBest(right, rank(best.R)+1)
40   if R exists then
41     push(q, Hyp(best.L, R))   // straight
42     push(q, Hyp(R, best.L))   // inverted
43   end
```

ses: a list of best hypotheses and a priority queue of candidates for the next best hypothesis. If the kth-best parse is already produced, it is the kth hypothesis in the best list. Otherwise, Cube Growing enumerates hypotheses by taking the best candidate from the priority queue until the kth hypothesis can be found. Whenever the best candidate is taken from the priority queue, successors of the candidate are pushed on the priority queue, if possible. To obtain the successors, Cube Growing is recursively performed.

[16] proposed that the original CYK parsing algorithm can be parallelized in three levels: sentence-level, cell-level, and grammar-level. Although they reported the grammar-level parallelization achieved the fastest result using thousands of GPUs, we adopted the cell-level parallelization. It is possible to parallelize the original CYK parsing at cell-level because the hypotheses in different chart cells covering same number of words in the sentence do not affect each other. Unfortunately, this property does not hold anymore in Cube Growing because k-best hypotheses are enumerated on demand. Therefore, a race condition arises if we directly apply Cube Growing in the cell-level parallelization.

We modified Cube Growing to fit in the cell-level parallelization. To avoid the race condition, the modified Cube Growing directly accesses to the priority queue for the first best hypothesis. It is postponed to move the first best hypothesis to the best list until the second best hypothesis is requested. From the second best parses, the modified Cube Growing does not run in parallel, which is identical to the original one. Algorithm 2 shows the entire procedures for the cell-level parallelization with the modified Cube Growing.

2.2.2. *Parallel feature generation*

The feature function ϕ in the discriminative model (Eq. 1) is further decomposed into the edge level in a reordering tree.

$$\phi_i(D, F) = \sum_{d \in D} \phi_i(d, F) \quad (2)$$

$$Score(D|F, \mathbf{w}) = \sum_{d \in D} \sum_i w_i \phi_i(d, F) \quad (3)$$

where d is a hyper edge in the hyper graph. Because most of feature functions $\phi_i(d, F)$ involve string operations, the feature generation becomes a major bottleneck of parsing efficiency. In a pilot study of our experiments, the feature generation is the most time-consuming process, which takes over 80% of the total parsing time.

Our proposed method parallelizes the feature generation in advance to produce a reordering tree. For a length-N sentence, there are $N(N-1)/2$ hyper edges in the hyper graph $\mathcal{D}$. For each hyper edge, there are two possible orientations $straight$ and $inverted$. Hence, the feature generation is performed $N(N-1)$ times in total.

With careful design of the feature function, the feature generation can be parallelized: If the feature function is defined only in a single level of the tree, in other words, a feature set generated from the feature function for a hyper edge is independent from that for the other edge. Therefore, two feature sets for two orientations are stored for each hyper edge, and thus $N(N-1)$ feature sets in the hyper graph in total.

Table 1: The statistics of corpora. Figures are the number of sentences. The first column shows the number of parallel sentences, and the second and third column show the numbers of monolingual sentences in German and English, respectively.

Data source	Parallel	German	English
WIT[3] [17]	138,499	146,206	158,641
Newswire	58,908	Not Used	
Europarl	2,399,123	Not Used	
Comman Crawl	1,920,209	Not Used	
News Commentary	178,221	204,276	247,966
News Crawl 2007	0	1,965,298	3,782,548
News Crawl 2008	0	6,690,332	12,954,477
News Crawl 2009	0	6,352,613	14,680,024
News Crawl 2010	0	2,899,914	6,797,225
News Crawl 2011	0	16,037,788	15,437,674
News Crawl 2012	0	20,673,844	14,869,673
Total	4,694,960	54,970,271	68,828,228

2.2.3. *On disk feature*

For each iteration, the feature sets generated by the feature function are identical, and the feature weights are only updated. To avoid redundant feature generation processes, reusing the generated features help the later iteration speed up. As the number of generated features is usually huge, however, it might be impossible store them in memory.

Our proposed method writes the features on disk after the generation instead of keeping them in memory. We simply create a file for each sentence with a identification of the sentence in the file name. At the actual parsing time, the generated features are recovered from the file for each sentence. Then the parser begins to search the best permutation π using the features according to the discriminative model. For each iteration, in other words, we skip the feature generation process and reuse the generated features at the first time.

3. Experimental result

In our experiments, we developed a reordering parser based on [11], `LADER`[1], and utilized a phrase-based SMT system `Moses` [13] for a reordering module and SMT module, respectively. The `tokenize.perl`[2] segmented German and English sentences into words. Word alignment of the segmented sentence pairs was performed using `MGIZA++` [18] for both German↔English directions, and refined using the

[1] `https://github.com/hwidongna/lader`

[2] `http://statmt.org/wmt08/scripts.tgz`

Table 2: The official evaluation results. XYZ in the first column refers the source X, the target Y and the priority of our run, where 1 is the primary and 2 is the contrastive. tst2013* denotes the results are measured on the reference with disfluency.

		Case-sensitive		Case-insensitive	
Run	Data	BLEU	TER	BLEU	TER
DE1	tst2013*	0.2126	0.6760	0.2174	0.6671
DE1	tst2013	0.2117	0.6890	0.2165	0.6804
ED1	tst2011	0.2348	0.5370	0.2406	0.5289
ED1	tst2012	0.2043	0.5913	0.2102	0.5805
ED1	tst2013	0.2243	0.5757	0.2300	0.5657
ED2	tst2011	0.2370	0.5337	0.2432	0.5256
ED2	tst2012	0.2036	0.5892	0.2105	0.5780
ED2	tst2013	0.2237	0.5764	0.2296	0.5665

grow-diag-final-and heuristics. A reordering parser utilized words and their automatically derived classes in the feature function. The training instances of the reordering parser were randomly selected among the word-aligned sentence pairs that licensed under ITG (around 3.5M sentences). For each iteration, the feature weights were updated using 10K instances according to Algorithm 1 and the maximum number of iterations was set to 100. We used the data supplied by the organizers of listed on the IWSLT 2013 Evaluation Campaign site. Table 1 summarizes the data statistics.

We submitted three runs: one for German-to-English (DE1) and two for English-to-German (ED1 and ED2). Our primary runs (DE1 and ED1) were the results of the preordering framework explained in Section 2. ED2 was a contrastive run using a hierarchical phrase-based SMT, which requires a longer time to decode. The decoding time of ED1 is almost half of ED2 excluding the reordering time. Table 2 shows the official results of the evaluation. The results from the other participant can be found in the overview paper [19].

Acknowledgement This work was partly supported by the IT R&D program of MSIP/KEIT (10041807), the CSLi corporation, the BK 21+ Project, and the National Korea Science and Engineering Foundation (KOSEF) (NRF-2009-0075211).

4. References

[1] Y. Zhang, R. Zens, and H. Ney, "Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation," in *Proceedings of SSST, NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, 2007, pp. 1–8.

[2] R. Tromble and J. Eisner, "Learning linear ordering problems for better translation," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 1007–1016.

[3] K. Visweswariah, R. Rajkumar, A. Gandhe, A. Ramanathan, and J. Navratil, "A word reordering model for improved machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 486–496.

[4] M. M. Khapra, A. Ramanathan, and K. Visweswariah, "Improving reordering performance using higher order and structural features," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 315–324. [Online]. Available: http://www.aclweb.org/anthology/N13-1032

[5] F. Xia and M. McCord, "Improving a statistical mt system with automatically learned rewrite patterns," in *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 2004, p. 508.

[6] M. Collins, P. Koehn, and I. Kučerová, "Clause restructuring for statistical machine translation," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 531–540.

[7] P. Xu, J. Kang, M. Ringgaard, and F. Och, "Using a dependency parser to improve smt for subject-object-verb languages," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009, pp. 245–253.

[8] D. Genzel, "Automatically learning source-side reordering rules for large scale machine translation," in *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 376–384.

[9] H. Isozaki, K. Sudoh, H. Tsukada, and K. Duh, "Head finalization: A simple reordering rule for sov languages," in *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Association for Computational Linguistics, 2010, pp. 244–251.

[10] J. DeNero and J. Uszkoreit, "Inducing sentence structure from parallel corpora for reordering," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11.

Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 193–203. [Online]. Available: http://dl.acm.org/citation.cfm?id=2145432.2145455

[11] G. Neubig, T. Watanabe, and S. Mori, "Inducing a discriminative parser to optimize machine translation reordering," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, ser. EMNLP-CoNLL '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 843–853. [Online]. Available: http://dl.acm.org/citation.cfm?id=2390948.2391039

[12] D. Wu, "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora," *Computational linguistics*, vol. 23, no. 3, pp. 377–403, 1997.

[13] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Annual meeting-association for computational linguistics*, vol. 45, 2007, p. 2.

[14] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, Dec. 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1248547.1248566

[15] L. Huang and D. Chiang, "Better k-best parsing," in *Proceedings of the Ninth International Workshop on Parsing Technology*, ser. Parsing '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 53–64. [Online]. Available: http://dl.acm.org/citation.cfm?id=1654494.1654500

[16] A. Dunlop, N. Bodenstab, and B. Roark, "Efficient matrix-encoded grammars and low latency parallelization strategies for cyk," in *Proceedings of the 12th International Conference on Parsing Technologies*, ser. IWPT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 163–174. [Online]. Available: http://dl.acm.org/citation.cfm?id=2206329.2206349

[17] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[18] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, ser. SETQA-NLP '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 49–57. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622110.1622119

[19] M. Cettolo, J. Niehues, S. Stker, L. Bentivogli, and M. Federico, "Report on the 10th iwslt evaluation campaign," in *Proceedings of the 10th International Workshop on Speech Language Translation*, 2013.

The RWTH Aachen Machine Translation Systems for IWSLT 2013

Joern Wuebker, Stephan Peitz, Tamer Alkhouli, Jan-Thorsten Peter
Minwei Feng, Markus Freitag and Hermann Ney

Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
Aachen, Germany
<surname>@cs.rwth-aachen.de

Abstract

This work describes the statistical machine translation (SMT) systems of RWTH Aachen University developed for the evaluation campaign *International Workshop on Spoken Language Translation* (IWSLT) 2013. We participated in the English→French, English↔German, Arabic→English, Chinese→English and Slovenian↔English MT tracks and the English→French and English→German SLT tracks. We apply phrase-based and hierarchical SMT decoders, which are augmented by state-of-the-art extensions. The novel techniques we experimentally evaluate include discriminative phrase training, a continuous space language model, a hierarchical reordering model, a word class language model, domain adaptation via data selection and system combination of standard and reverse order models. By application of these methods we can show considerable improvements over the respective baseline systems.

1. Introduction

We describe the statistical machine translation (SMT) systems developed by RWTH Aachen University for the evaluation campaign of IWSLT 2013. We participated in the machine translation (MT) track for the language pairs English→French, English↔German, Arabic→English, Chinese→English and Slovenian↔English and the spoken language translation (SLT) tracks for the language pairs English→French and English→German. We apply state-of-the-art phrase-based and hierarchical machine translation systems as well as an in-house system combination framework. To improve the baselines, we evaluated several different methods in terms of translation performance. These include a discriminative phrase training technique, continuous space language models, a hierarchical reordering model for the phrasal decoder, word class (cluster) language models, domain adaptation via data selection, application of two separate translation models or phrase table interpolation, word class translation and reordering models, optimization with PRO and a discriminative word lexicon. Further, on the small scale Slovenian↔English tasks we compare the performance of the two word alignment toolkits GIZA++ and *fast_align*. For the spoken language translation task, the ASR output is enriched with punctuation and casing. The enrichment is performed by a hierarchical phrase-based translation system.

This paper is organized as follows. In Section 2 we describe our translation software and baseline setups. Sections 2.3 and 2.4 introduce the novel discriminative phrase training technique and the continuous space language model, whose application shows improvements on several tasks. Our experiments for each track are summarized in Section 3 and we conclude with Section 4.

2. SMT Systems

For the IWSLT 2013 evaluation campaign, RWTH utilized state-of-the-art phrase-based and hierarchical translation systems as well as our in-house system combination framework. GIZA++ [1] or *fast_align* [2] are employed to train word alignments. All language models are created with the SRILM toolkit [3] and are standard 4-gram LMs with interpolated modified Kneser-Ney smoothing. We evaluate in case-insensitive fashion, using the BLEU [4] and TER [5] measures.

2.1. Phrase-based Systems

As phrase-based SMT systems, in this work we used both an in-house implementation of the state-of-the-art MT decoder (PBT) described in [6] and the implementation of the decoder based on [7] (SCSS) which is part of RWTH's open-source SMT toolkit Jane 2.1 [1]. We use the standard set of models with phrase translation probabilities and lexical smoothing in both directions, word and phrase penalty, distance-based reordering model, an n-gram target language model and three binary count features. The parameter weights are optimized with MERT [8], PRO [9] (SCSS) or the downhill simplex algorithm [10] (PBT).

Additional state-of-the-art models that are applied successfully in the IWSLT 2013 evaluation are a hierarchi-

[1] http://www-i6.informatik.rwth-aachen.de/jane/

cal reordering model (HRM) [11], a high-order word class language model (wcLM) [12], word class based translation and reordering models (wcTM) [12], a discriminative phrase training scheme (cf. Section 2.3) and rescoring with a neural network language model (cf. Section 2.4).

2.2. Hierarchical Phrase-based System

For our hierarchical setups, we employed the open source translation toolkit Jane [13], which has been developed at RWTH and is freely available for non-commercial use. In hierarchical phrase-based translation [14], a weighted synchronous context-free grammar is induced from parallel text. In addition to contiguous *lexical* phrases, *hierarchical* phrases with up to two gaps are extracted. The search is carried out with a parsing-based procedure. The standard models integrated into our Jane systems are: phrase translation probabilities and lexical smoothing probabilities in both translation directions, word and phrase penalty, binary features marking hierarchical phrases, glue rule, and rules with non-terminals at the boundaries, four binary count features, phrase length ratios and an n-gram language model. We utilize the cube pruning algorithm [15] for decoding and optimize the model weights with standard MERT [8] on 100-best lists.

2.3. Discriminative Phrase Training

The state of the art for creating the phrase tables of standard SMT systems is still a heuristic extraction from word alignments and probability estimation as relative frequencies. In several systems for the IWSLT 2013 shared task, we applied a more sophisticated discriminative phrase training method. Similar to [16], a gradient-based method is used to optimize a maximum expected BLEU objective, for which we define BLEU on the sentence level with smoothed 3-gram and 4-gram precisions. In the experiments reported in this paper, we perform discriminative training on the TED portion of the training data in all cases. To that end, we decode the training data to generate 100-best lists. A leave-one-out heuristic [17] is applied to make better use of the training data. Using these n-best lists, we iteratively perform updates on the phrasal translation scores of the phrase table. After each iteration, we perform MERT, evaluate on the development set and finally select the iteration which performs best.

2.4. Neural Network Language Model

We train neural networks as language models using the theano numerical computation library[18]. The neural network structure is largely similar to the continuous space language model (*CSLM*) [19]. Our input layer includes a short list of the most common word and word factors like the word beginning or ending. To reduce the computation cost of the network we employ a clustered output layer [20, 21]. The Neural Network Language Model is used as a final step in our translation pipeline, by rescoring on 200-best lists for the English→French and English→German tasks.

Table 1: Results for the English→French MT task.

system	dev2010		tst2010	
	BLEU	TER	BLEU	TER
SCSS allData	28.3	55.7	31.9	49.8
+HRM	28.7	55.3	32.5	49.2
+2TM	29.2	54.7	32.7	48.9
+GW	29.5	54.6	32.9	48.9
+DWL	29.8	54.3	33.2	48.5
+wcLM	29.7	54.2	33.5	48.3
+CSLM	30.0	53.8	33.7	48.0

3. Experimental Evaluation

3.1. English→French

For the English→French task, the word alignment was trained with GIZA++ and we applied the phrase-based decoder implemented in Jane. We used all available parallel data for training the translation model. The baseline French LM is trained on the target side of all available bilingual data plus $\frac{1}{2}$ of the Shuffled News corpus. The monolingual data selection is based on cross-entropy difference as described in [22]. The experimental results are given in Table 1. Different from last year [23], we did not employ system combination in this task, achieving similar results with a single decoder. The baseline system is improved by the hierarchical reordering model (HRM, +0.6% BLEU), adding a second translation model to the decoder (2TM, +0.2% BLEU), which was trained on the TED portion of the data, using $\frac{1}{4}$ of the French Gigaword Second Edition corpus as additional language model training data (GW, +0.2% BLEU), and smoothing the translation model with a discriminative word lexicon [24] trained on the in-domain data (+0.3% BLEU). For the final submission, we applied two additional language models: the 7-gram word class language model (wcLM, 0.3% BLEU) and the neural language model (CSLM, 0.2% BLEU).

3.2. German↔English

Similar to English→French, for the German↔English tasks, we used GIZA++ for the word alignments and applied the phrase-based decoder from the Jane toolkit.

For the German→English translation direction, in a preprocessing step the German source is decompounded [25] and part-of-speech-based long-range verb reordering rules [26] are applied. The English LM is trained the target side of all available bilingual data plus a selection [22] of $\frac{1}{2}$ from the Shuffled News corpus and $\frac{1}{4}$ from the English Gigaword v3 corpus, resuling in a total of 1.7 billion running words. The experimental results for the German→English task are given in Table 2. In opposition to our findings from last

Table 2: Results for the German→English MT task.

system	dev2010		tst2010	
	BLEU	TER	BLEU	TER
SCSS TED	31.5	47.6	30.0	49.2
SCSS allData	32.8	46.4	30.3	48.9
+HRM	33.0	46.1	30.4	48.9
+wcLM	33.5	45.8	30.9	48.4
+discr.	33.9	45.0	31.4	47.5
+2TM	34.2	45.2	32.3	47.4

Table 3: Results for the English→German MT task.

system	dev2010		tst2010	
	BLEU	TER	BLEU	TER
SCSS TED	22.0	56.7	21.9	57.3
SCSS allData	22.7	56.1	22.3	57.2
+HRM	23.3	55.5	22.6	57.7
+wcLM	24.2	54.5	23.6	55.9
+discr.	24.6	54.1	24.3	55.4
+CSLM	24.7	53.7	24.9	54.7

year [23], using all available data now performs better than solely training on the in-domain TED portion. This can be attributed to the large, newly available Common Crawl corpus. The baseline system is improved by the hierarchical reordering model (HRM, +0.1% BLEU), the 7-gram word class language model (wcLM, 0.5% BLEU) and discriminative phrase training (discr., +0.5% BLEU). Finally, we applied domain adaptation by adding a second translation model to the decoder (2TM), which was trained on the TED portion of the data. This second translation model was also trained with discriminative phrase training and gave an additional improvement of 0.9% BLEU.

The English→German system is very similar to the one for the opposite translation direction. The language model was trained on the target side of all bilingual data plus $\frac{1}{2}$ of the Shuffled News corpus selected with [22]. The LM training data contains a total of 564 million running words. The results in Table 3 show that using all available training data outperforms only training on the in-domain TED portion. The system is augmented with the hierarchical reordering model (HRM, +0.3% BLEU), a word class language model (wcLM, 1.0% BLEU) and disriminative phrase training (discr., +0.5% BLEU). Especially the wcLM has a strong impact on translation performance. Different from the opposite direction, adding a second translation model did not improve results. However, we were able to reach a final improvement of 0.6% BLEU by rescoring a 200-best list with a neural language model (CSLM).

3.3. Arabic→English

The Arabic→English system uses a language model based on the full in-domain TED and out-of-domain UN and News Commentary v8 data. We also filtered and included the English Gigaword, giga-fren.en, Europarl v7, Common Crawl and Shuffled News corpora using the cross-entropy criterion. A 4-gram LM is trained for each of the sets using modified Kneser-Ney discounting with interpolation. The final LM is the weighted mixture of all individual LMs, with the weights tuned to achieve the lowest perplexity on dev2010. We also trained another mixture of LMs keeping singleton *n*-grams, which we will refer to as sngLM.

A single system employing MADA v3.1 D3 resulted in only 0.3% worse BLEU and TER on the tst2011 dataset of IWSLT2012, compared to a system combination where single systems of various segmentation techniques were combined, as described in [23]. Therefore, we stuck to a single system using MADA v3.1 D3 for segmentation. The translation model is trained using the TED and UN bilingual corpora, and the standard features were used in addition to HRM. Two phrase tables were built, one based on the TED dataset and the other on the TED+UN data. We interpolated the two linearly with the weights 0.9 and 0.1 respectively given to the TED and the full phrase tables. Table 4 shows the results. The HRM features bring an improvement of 1.1% BLEU and 0.2% TER to a TED-only translation model. Adding the UN data hurts performance by 1.1% BLEU and 0.7% TER. On the other hand, interpolation leads to an improvement of 0.8% in TER and 0.1% BLEU. When replacing the LM with sngLM an improvement is only observed on the development set, but not the test set, which could not be remedied by relaxing the pruning parameters. All sngLM experiments used a 200-best list, compared to a 100-best list used with the smaller LM.

We also experimented with bilingual filtering of the UN data used to train the phrase table, where scoring was performed using bilingual LM cross-entropy scores (x-entropy) [27]. Another experiment used the combination of cross-entropy and IBM-1 scores (x-entropy+IBM-1) [28]. We used the best 400k UN sentences together with the TED data to train a phrase table, which is then interpolated with a TED-only phrase table as described above. x-entropy+IBM-1 is better by 0.8% TER than mere cross-entropy filtering, and it performs similar to the non-filtered system, despite the fact that we select only $\frac{1}{16}$ of the UN data.

3.4. Chinese→English

For the Chinese-English task, RWTH utilized system combination as described in [29]. We used both the phrase-based decoder and the hierarchical phrase-based decoder to perform a bi-directional translation, which means the system performs standard direction decoding (left-to-right) and reverse direction decoding (right-to-left). To build the reverse direction system, we used exactly the same data as the stan-

Table 4: Results for the Arabic→English MT task.

system	dev2010		tst2010	
	BLEU	TER	BLEU	TER
SCSS TED	27.4	52.0	25.7	55.1
+HRM	27.9	51.9	26.8	54.9
+UN	28.4	51.9	25.7	55.6
+UN interpolated	28.3	51.1	26.9	54.1
+sngLM	28.8	50.7	26.8	54.1
+x-entropy	28.6	51.8	26.7	55.0
+x-entropy+IBM-1	28.8	51.0	27.0	54.2

Table 5: Chinese-English results on the dev test set for different segmentations. The primary submission is a system combination of all the listed systems.

system	dev2010		tst2010	
	BLEU	TER	BLEU	TER
PBT-2012-standard	11.5	80.7	13.0	76.4
PBT-2012-reverse	11.7	80.9	13.6	75.5
HPBT-2012-standard	12.3	79.8	14.2	74.6
HPBT-2012-reverse	12.8	79.4	14.6	74.1
HPBT-2013-standard	12.4	79.5	14.5	74.1
HPBT-2013-reverse	12.6	79.4	14.4	74.3
system combination	13.5	78.5	15.1	73.6

dard direction system and simply reversed the word order of the bilingual corpora. For the system combination we selected four systems we had trained for last year's IWSLT evaluation and set up two additional hierarchical systems with slightly different preprocessing. Note that all translation model are trained on the in-domain data only. By performing system combination we gain an improvement of +0.5% BLEU over the best single system. Results are given in Table 5.

3.5. Slovenian↔English

The bilingual training data available for the Slovenian↔English tasks is limited to 14K sentence pairs from the TED lecture domain. Further, only one development set was provided. In order to be able to do blind evaluation, we split it into two parts. The first 644 lines are defined as `dev1` and are used for MERT/PRO. The remaining 500 lines are used as blind test set and will be referred to as `dev2`. For the Slovenian↔English tasks, we apply our phrase-based decoder and experimented with two different word alignments for training, one generated with GIZA++, based on the IBM model 4, and one created with *fast_align*, which uses a reparameterization of IBM model 2. Interestingly, the simpler and more efficient *fast_align* tool outperforms GIZA++ in both cases.

Table 6: Results for the Slovenian→English MT task. All systems are augmented with the hierarchical reordering model.

system	dev1		dev2	
	BLEU	TER	BLEU	TER
SCSS GIZA++	17.6	65.7	15.9	67.6
SCSS *fast_align*	18.0	64.8	16.3	66.1
+wcLM	18.2	62.9	16.5	64.6
+wcTM +PRO	18.6	63.0	16.5	64.3
+discr.	18.8	62.6	16.9	63.9

Table 7: Results for the English→Slovenian MT task. All systems are augmented with the hierarchical reordering model.

system	dev1		dev2	
	BLEU	TER	BLEU	TER
SCSS GIZA++	11.3	70.5	9.6	71.4
SCSS *fast_align*	11.4	70.3	10.5	69.6
+wcLM	12.0	69.8	10.1	69.9
+wcTM	11.9	70.3	10.4	69.9
+discr.	11.9	70.2	10.7	69.7

The Slovenian→English MT system uses the same language model as described in Section 3.2 for the German→English task. Results are shown in Table 6. The baseline, which already contains the hierarchical reordering model, is augmented with a word class LM (wcLM, +0.2% BLEU) and the word class translation and reordering model (wcTM). When we add the latter, we switch from MERT to PRO, which we found to lead to more stable results in this case. Finally, we employ discriminative phrase training (discrim., +0.4% BLEU) to build the submission system.

To train the Slovenian language model only the target side of the bilingual data was provided. We found that selecting a submission system on this task was very difficult, as when comparing two setups, their behaviour was often reversed between `dev1` and `dev2`. We decided to apply the same extensions to the baseline as for the opposite translation direction. The baseline, which already contains the hierarchical reordering model, is augmented with the word class LM and the word class based translation and reordering models. Here, we continue using MERT. For the final submission, we also applied discriminative phrase training. Results are shown in Table 7.

3.6. Spoken Language Translation (SLT)

RWTH participated in the English→French and English→German SLT task. In both tracks, we reintroduced punctuation and case information following [30],

Table 8: Results for the English→French SLT task.

system	dev2010		tst2010	
	BLEU	TER	BLEU	TER
	23.0	62.7	26.0	56.0
re-optimized	23.4	62.5	26.3	56.2

which we denote as *enriched*. Further, we added a phrase feature, that fires if a phrase introduces a punctuation mark on the target side. The SMT system, that is employed in the enrichment process by translating from pure ASR output to the enriched version, we use a hierarchical phrase-based system with a maximum of one nonterminal symbol per rule The model weights are tuned with standard MERT on 100-best lists. As optimization criterion we use WER.

For English→French, we re-optimized on the enriched ASR development dev using **SCSS allData** +HRM +GW +2TM. Results are reported in Table 8.

For English→German, the enriched evaluation set was translated using the **SCSS allData** +HRM +wcLM +discr. system. Here, the translation system was kept completely unchanged from the MT task, including the log-linear feature weights.

4. Conclusion

RWTH participated in seven MT tracks and two SLT tracks of the IWSLT 2013 evaluation campaign. The baseline systems utilize our state-of-the-art translation decoders and we were able to improve them by applying novel models or techniques. The most notable improvements are achieved by a hierarchical reordering model (+1.1 BLEU on Ar-En), a word class language model (+1.0 BLEU on En-De), discriminative phrase training (+0.7 BLEU on En-De), a continuous space language model (+0.6 BLEU on En-De) and system combination of standard and reverse order models (+0.5 BLEU on Zh-En). For the SLT track, the ASR output was enriched with punctuation and casing information by a hierarchical translation system.

5. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreements n° 287658 and n° 287755. This work was also partly achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

6. References

[1] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[2] C. Dyer, V. Chahuneau, and N. A. Smith, "A Simple, Fast, and Effective Reparameterization of IBM Model 2," in *Proceedings of NAACL-HLT*, Atlanta, Georgia, June 2013, pp. 644—-648.

[3] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *Proc. of the Int. Conf. on Speech and Language Processing (ICSLP)*, vol. 2, Denver, CO, Sept. 2002, pp. 901–904.

[4] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, July 2002, pp. 311–318.

[5] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA, August 2006, pp. 223–231.

[6] R. Zens and H. Ney, "Improvements in Dynamic Programming Beam Search for Phrase-based Statistical Machine Translation," in *International Workshop on Spoken Language Translation*, Honolulu, Hawaii, Oct. 2008, pp. 195–205.

[7] J. Wuebker, M. Huck, S. Peitz, M. Nuhn, M. Freitag, J.-T. Peter, S. Mansour, and H. Ney, "Jane 2: Open source phrase-based and hierarchical statistical machine translation," in *International Conference on Computational Linguistics*, Mumbai, India, Dec. 2012, to appear.

[8] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003, pp. 160–167.

[9] M. Hopkins and J. May, "Tuning as ranking," in *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, July 2011, pp. 1352–1362.

[10] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, pp. 308–313, 1965.

[11] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 848–856. [Online]. Available: http://dl.acm.org/citation.cfm?id=1613715.1613824

[12] J. Wuebker, S. Peitz, F. Rietig, and H. Ney, "Improving statistical machine translation with word class models," in *Conference on Empirical Methods in Natural Language Processing*, Seattle, USA, Oct. 2013, pp. 1377–1381.

[13] D. Vilar, D. Stein, M. Huck, and H. Ney, "Jane: Open source hierarchical translation, extended with reordering and lexicon models," in *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden, July 2010, pp. 262–270.

[14] D. Chiang, "Hierarchical Phrase-Based Translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[15] L. Huang and D. Chiang, "Forest Rescoring: Faster Decoding with Integrated Language Models," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, June 2007, pp. 144–151.

[16] X. He and L. Deng, "Maximum Expected BLEU Training of Phrase and Lexicon Translation Models," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, Jeju, Republic of Korea, Jul 2012, pp. 292–301.

[17] J. Wuebker, A. Mauser, and H. Ney, "Training phrase translation models with leaving-one-out," in *Proceedings of the 48th Annual Meeting of the Assoc. for Computational Linguistics*, Uppsala, Sweden, July 2010, pp. 475–484.

[18] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010, oral Presentation.

[19] H. Schwenk, A. Rousseau, and M. Attik, "Large, Pruned or Continuous Space Language Models on a GPU for Statistical Machine Translation," in *NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, Montréal, Canada, June 2012, pp. 11–19.

[20] J. Goodman, "Classes for fast maximum entropy training," *CoRR*, vol. cs.CL/0108006, 2001.

[21] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, R. G. Cowell and Z. Ghahramani, Eds. Society for Artificial Intelligence and Statistics, 2005, pp. 246–252. [Online]. Available: http://www.iro.umontreal.ca/~lisa/pointeurs/hierarchical-nnlm-aistats05.pdf

[22] R. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *ACL (Short Papers)*, Uppsala, Sweden, July 2010, pp. 220–224.

[23] S. Peitz, S. Mansour, M. Freitag, M. Feng, M. Huck, J. Wuebker, M. Nuhn, M. Nußbaum-Thom, and H. Ney, "The rwth aachen speech recognition and machine translation system for iwslt 2012," in *International Workshop on Spoken Language Translation*, Hong Kong, Dec. 2012, pp. 69–76. [Online]. Available: http://hltc.cs.ust.hk/iwslt/proceedings/paper_45.pdf

[24] A. Mauser, S. Hasan, and H. Ney, "Extending statistical machine translation with discriminative and trigger-based lexicon models," in *Conference on Empirical Methods in Natural Language Processing*, Singapore, Aug. 2009, pp. 210–217.

[25] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *Proceedings of European Chapter of the ACL (EACL 2009)*, 2003, pp. 187–194.

[26] M. Popović and H. Ney, "POS-based Word Reorderings for Statistical Machine Translation," in *International Conference on Language Resources and Evaluation*, 2006, pp. 1278–1283.

[27] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., July 2011, pp. 355–362.

[28] S. Mansour, J. Wuebker, and H. Ney, "Combining Translation and Language Model Scoring for Domain-Specific Data Filtering," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, Dec. 2011.

[29] M. Freitag, M. Feng, M. Huck, S. Peitz, and H. Ney, "Reverse word order models," in *Machine Translation Summit*, Nice, France, Sept. 2013.

[30] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling Punctuation Prediction as Machine Translation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, Dec. 2011.

Description of the UEDIN System for German ASR

Joris Driesen, Peter Bell, Mark Sinclair, Steve Renals

Center for Speech Technology Research, University of Edinburgh, UK

{jdriesen,peter.bell,s.renals}@inf.ed.ac.uk, M.Sinclair-7@sms.ed.ac.uk

Abstract

In this paper we describe the ASR system for German built at the University of Edinburgh (UEDIN) for the 2013 IWSLT evaluation campaign. For ASR, the major challenge to overcome, was to find suitable acoustic training data. Due to the lack of expertly transcribed acoustic speech data for German, acoustic model training had to be performed on publicly available data crawled from the internet. For evaluation, lack of a manual segmentation into utterances was handled in two different ways: by generating an automatic segmentation, and by treating entire input files as a single segment. Demonstrating the latter method is superior in the current task, we obtained a WER of 28.16% on the dev set and 36.21% on the test set.

Index Terms: Light supervision, Segmentation, Acoustic Model Training

1. Introduction

In ASR, good acoustic models are an important prerequisite for high recognition accuracies. The quality of these models is determined by both the quality and the quantity of the data on which they were trained. Such data consists of speech as well as accurate orthographic transcriptions. Since the latter must be manually created by human transcribers, which is a slow and expensive process, it can be difficult to obtain training data in sufficiently large quantities. In languages or domains where resources are scarce, i.e., where no large amounts of dedicated transcribed training is available, acoustic models can still be obtained from untranscribed or poorly transcribed data, using unsupervised or lightly supervised training methods [1, 2, 3, 4, 5]. Since German ASR has historically received little attention at UEDIN, there are very few resources available for it on site. Therefore, even though German is by no means an under-resourced language, we have been compelled to treat it as such, collecting large amounts of publicly available data and processing it with the lightly supervised training methods mentioned above. Although this methodology is not strictly necessary for German, it can in theory be applied to unlock other, truly under-resourced languages, for which no alternative training methods exist. The available resources used for acoustic model training are discussed below in section 2. The lightly supervised training is explained fully in section 4. Acoustic model training is finalised by training a Deep Neural Network (DNN) in a hybryd setup with a traditional context-dependent tri-phone based Hidden Markov Model (HMM), as explained below, in section 6.

Aside from acoustic modelling, the proposed system has state-of-the-art language modelling. In a first phase, text corpora are collected, containing in total almost 10^9 words. Based on the cross-entropy with the evaluation domain, as proposed in [6], the top 30 percentile of this data is selected and 4-gram language models, as well as Recurrent Neural Network Language Models (RNNLM) are trained on it [7]. Details of this setup can be found below, in section 5.

Since no manual segmentation for the evaluation set is provided, it is necessary to produce a segmentation automatically. Alternatively, ASR can be performed on entire talks, treating them as a single segment. There is an inherent trade-off between these approaches, since each has its own advantages and disadvantages. A segmentation that is generated automatically may contain erroneous segment boundaries, which can easily lead to recognition errors. When segmentation is avoided, on the other hand, recognition could be performed on non-speech segments, generating unpredictable erroneous outputs. In section 6, evaluation is performed comparing both approaches.

2. Available Resources for Acoustic Modelling

The data on which an ASR system is trained determines to a large extent its eventual performance. Several properties of the training data are important. Firstly, its domain must be matched as closely as possible to the domain of the evaluation set. Even when using techniques like fMLLR [8] to adapt acoustic models to the test domain, any mismatch will significantly reduce recognition accuracies. Also accurate orthographic transcriptions of the training data are necessary. Even small amounts of transcription errors can significantly reduce recognition performance, e.g. [9]. Lastly, the size of the training set plays an important role. Although there is no such thing as a direct linear relation between training set size and recognition performance, having more training data does usually lead to better results. Several tens of hours is believed to be a minimum for acoustic model training, depending on

This work has been funded by the European Union as part of the Seventh Framework Programme, under grant agreement no. 287658 (EU-BRIDGE), and by EPSRC Programme Grant grant EP/I031022/1, *Natural Speech Technology*.

the size and complexity of the models being trained.

2.1. Globalphone

One of the suitable speech corpora accessible to us is GlobalPhone [10]. It is a multi-lingual corpus, covering a selection of the world's most widely spoken languages, one of which is German. For each language, it contains speech from about 100 adult native speakers, reading a number of articles taken from a local newspaper. For German, this adds up to about 18 hours of speech. Only 14 hours of this can be used as training data, since the rest is divided over a dev set and a test set. In the context of this paper, the GlobalPhone corpus is less suitable for acoustic model training, due to its small size and its large domain mismatch with the IWSLT evaluation data. However, the German lexicon that is included in the corpus is invaluable to us, since it is the only lexicon we have at our disposal. It contains 36994 unique words, with 39520 pronunciations, indicating that a relatively large number of words is listed with more than a single pronunciation variant. Furthermore, a 3-gram language model for this data is available to us. It is the same language model that was used in [11], and is specifically tuned to the domain of news articles. Using this LM is not our only option though, since we have the option to train our own, more tuned to the domain of TED-talks, see section 5.

2.2. Europarl

The second set of data was obtained by crawling the website of the European Parliament [12], which has committed itself to making its plenary sessions publicly available online, along with their transcripts. These sessions contain speech in a wide variety of languages, German among them. Although, generally speaking, the transcriptions do not match the spoken content of the speech perfectly, techniques for lightly supervised acoustic model training may be employed to circumvent this. We will elaborate on this below in section 4. In this work, we downloaded all parliamentary sessions of the years 2008, 2009, and 2010. This is about 990 hours of audio data. This data contains 23 audio streams in parallel: one stream with the raw unaltered recordings, and one additional stream for each of the 22 languages of the European Union. In these audio streams, speech in any other language than the target language is replaced with its on-the-fly translation, done in real-time by professional interpreters. For each parliamentary speech, there is only a single start and end time given, shared over all 22 parallel versions of that speech. Since translations may take longer than the original speech, or may be shifted in time, the audio segments delineated by these boundaries are usually 10–20 seconds longer than the speech they contain, and tend to overlap each other. Adding the lengths of all these segments together therefore leads to an overestimate of the available data, but can nonetheless be a useful indication. The total amount of speech data we counted like this, is 733 hours. One must be cautious in using all this data directly, however, since it contains directly recorded speech from German-speaking MEP's, as well as interpreters' speech. There are very distinct differences between these types of speech: e.g. whereas MEP's speak more spontaneously, often with an accent, interpreters tend to speak clearly, with long pauses, and very few corrections and repetitions. Since these types of speech may not be equally well matched to the target domain, we have treated them separately. We identified the speeches that were originally spoken in German, by comparing the German audio stream with the raw unaltered audio. Based on the same rough count as before, this adds up to about 95 hours of speech. Since there is no lexicon available with this data, we reuse the GlobalPhone lexicon, to which the out-of-vocabulary words are added using Sequitor Grapheme-to-Phoneme conversion [13].

3. Text Tokenisation

Although the GlobalPhone lexicon does contain 373 numbers, this list is far from exhaustive. Numbers in the evaluation data are therefore very likely to be OOV. To prevent this from happening, we defined rewrite rules to convert any number that is OOV into its constituent parts, most of which do occur in the lexicon, or are easily added to it. For instance, if "1,234" is encountered, it is rewritten as "1,000 2 100 4 und 30". This way, with no more than 33 lexical entries, we are able to handle any number between 1 and $9,999 \cdot 10^6$. Special exception rules are provided to deal with such things as times, dates, years, and IP-addresses. Measures of distance, length, and volume are fully expanded, as well as currencies, e.g. 'km' is written as 'kilometer', '$' is written as 'dollar', etc. Because of time constraints, handling of abbreviations in our system is rudimentary. Basically, any word that either consists of two or more capitalized letters, or of letters separated by full stops is recognized as an abbreviation. They are then written in a consistent form, namely as uncapitalized letters separated by full stops, and then added to the lexicon using grapheme-to-phoneme conversion. There are several ways in which this methodology is suboptimal. For one, it disregards the possibility of abbreviations being pronounced as words, rather than sequences of separate letters, e.g. the pronunciation of "NATO" as /nato/ rather than /ɛnɑteo/. More importantly, the GlobalPhone lexicon, on which we trained the grapheme-to-phoneme conversion, contains far too few examples to enable accurate pronunciation predictions. As a result, abbreviations in training and evaluation data are expected to reduce the performance of our system.

4. Lightly Supervised Acoustic Model Training

To perform acoustic model training and evaluation, the acoustic data is preprocessed as follows. First, it is converted towards mono-channel 16kHz WAVE-files. MFCC-

coefficients are determined within 25 ms frames which are shifted in increments of 10 ms. Cepstral Mean Normalisation is then applied to the resulting 13-dimensional feature vectors. For each frame, the features within a context window of 9 frames, 4 to the left, 4 to the right, are stacked and projected down to 39 dimensions using LDA-MLLT.

4.1. Training an Initial Model on GlobalPhone

We train an initial GMM-HMM acoustic model from scratch on the GlobalPhone corpus. This model contains 3000 context-dependent states and 48000 Gaussians. It was evaluated on three different evaluation sets: the GlobalPhone dev set, where it resulted in a WER of 12.68%, the GlobalPhone eval set, on which it gave a WER of 19.92%, and the IWSLT dev set, on which it yielded a WER of 56.18%. The language model used in each of these evaluations was the GlobalPhone-specific one, introduced in section 2.1.

4.2. Further Training on Europarl

Acoustic model training on Europarl data cannot be done straightforwardly, since the transcriptions we have of it do not match the acoustics perfectly. There is a variety of light supervision techniques, however, with which this problem may be circumvented, e.g. [14, 1]. Here, we used the greedy matching approach described in [5]. We first bias the GlobalPhone LM towards the Europarl domain by interpolating it with a small LM trained on the imperfect transcriptions. This LM, in combination with the acoustic model trained above in section 4.1, is then used to make a recognition of the Europarl training data. By comparing the recognition result with the imperfect transcription, and greedily collecting the longest sequences that occur in both, a new in-domain training set is constructed. From this, a new acoustic model with the same number of states and Gaussians is trained and the whole process is repeated. This iterative process is illustrated in figures 1 and 2. With each iteration, the accuracy of the ASR transcription is expected to rise, and hence more training data is collected for the iteration after that. Also, with each iteration, the models are expected to get more tuned towards the Europarl domain. In this work, we first apply this technique for 10 iterations on the subset with 95 hours of direct MEP recordings, discussed in section 2.2, and evaluated on the IWSLT dev set in each iteration. The result is shown in the leftmost columns of table 1. The initial WER of 46.36% is obtained with the GlobalPhone acoustic model. The reason why this result is different from the 56.18% reported in section 4.1 is that another LM was used in these evaluations, namely the one that is biased towards Europarl data. Looking at the WER's, we can see that the quality of the acoustic models doesn't improve with each new iteration. If anything, the opposite is true, although the statistical significance of these differences may be questionable. This lack of improvement is probably caused by a slight domain mismatch between Europarl and the TED talks in the IWSLT dev set. An interesting experiment would be to evaluate the models in each iteration on an evaluation set in the Europarl domain. Unfortunately, no such evaluation set is available to us. When doing the same experiment on the entire Europarl corpus, MEP speech and interpreters' speech put together, the results become as shown in the rightmost columns of table 1. The acoustic model obtained in iteration 10 of the previous experiment is used here as the initial acoustic model. Although the WER drops about 1% absolute with the inclusion of the interpreters' speech, the results are otherwise comparable to those of the previous experiment. The drop in WER is very likely due to the increase of the training set from 46.98 hours to 67.18 hours. The best performance, a WER of 39.95%, is achieved in the third iteration. Therefore, the training set obtained in that iteration is used for all acoustic model training in further experiments, see section 6.

	MEP		All	
iter	hours	WER(%)	hours	WER(%)
init	NA	46.36	46.98	41.12
1	45.91	41.13	67.15	40.22
2	46.64	41.20	70.28	40.09
3	46.69	41.36	70.80	**39.95**
4	46.80	41.25	70.83	40.01
5	46.89	41.10	70.92	40.27
6	47.00	41.36	70.93	40.28
7	47.07	41.55	70.99	40.26
8	47.01	41.49	70.95	40.12
9	47.00	41.28	70.89	40.50
10	46.98	41.12	70.94	40.35

Table 1: *The data set sizes and WER rates obtained on the IWSLT dev set in each iteration of lightly supervised training.*

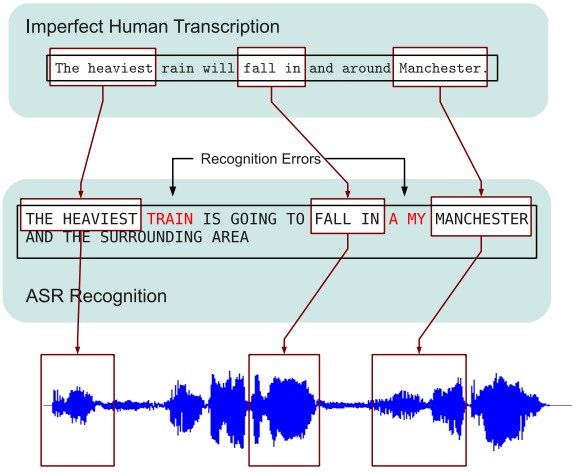


Figure 1: *The longest word sequences occurring both in the approximate transcription and in the ASR output are identified.*

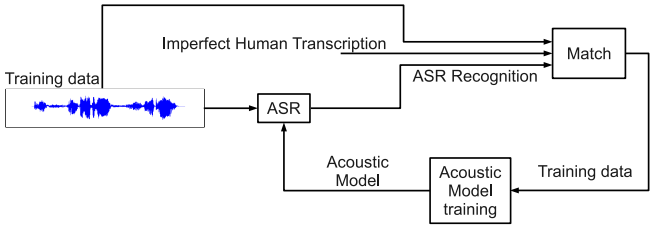


Figure 2: *Illustration of the iterative process, in which training data is collected to obtain acoustic models, which are in turn used to collect a better set of training data.*

name	# words ($\cdot 10^6$)
europarl_v7	47.37
europarl_crawl	2.86
news_crawl_2007	31.47
news_crawl_2008	107.86
news_crawl_2009	101.56
news_crawl_2010	45.89
news_crawl_2011	252.85
news_crawl_2012	319.73
news_comment	4.45
total	914.05

Table 2: *The text resources used for LM training.*

5. Language Modelling

For language model training, we used the resources listed in table 2. All of these were obtained through links on the IWSLT website, except 'europarl_crawl', which consists of the imperfect transcriptions of the Europarl data from section 2.2. All text was first depunctuated and tokenised as described in section 3. From each of these texts, 30% is selected that best matches the domain of the IWSLT dev set, according to the cross-entropy criterion proposed in [6]. Language models are trained on this subset only, disregarding the remaining 70%.

5.1. N-gram Language Models

After winnowing them down to 30%, each of the text corpora is used to train a 3-gram LM, using the MITLM language modelling toolkit [15]. In this training, modified Kneser-Ney smoothing [16] is used with parameters optimised on the IWSLT dev set. These language models are then linearly interpolated with interpolation weights optimised in the same way. The 1-grams in the resulting interpolated model are then written out in decreasing order, according to their smoothed 1-gram probability. Choosing the top-N words from this list allows us to optimally define a dictionary of size N for further LM training. We then repeated the previous procedure, training 3-gram LM's on the whittled down text corpora, with a limited vocabulary of N words, and linearly interpolating them. Finally the same was done with

N	OOV rate(%)	3-gram ppl.	4-gram ppl.
100000	4.18	252.63	246.36
150000	3.32	278.24	263.37
200000	2.78	283.25	275.97
250000	2.52	289.73	282.43
300000	2.37	294.24	**286.86**
350000	2.29	297.03	289.74
400000	2.17	300.30	292.97

Table 3: *The perplexities and OOV rates of the 3-gram and 4-gram LM's on the IWSLT dev set*

4-grams. The OOV rate and perplexity on the dev set for a range of values for N is shown in table 3. As expected, the 4-gram models achieve lower perplexities than 3-gram models. Based on these results, we choose the 4-gram LM with vocabulary size 300000 for the evaluations in section 6, since this yields a good trade-off between word coverage and perplexity. Any of these 300000 words that do not occur either in GlobalPhone or in the crawled Europarl data is added to the lexicon. Using a LM of such size for LVCSR (Large Vocabulary Continuous Speech Recognition) is very demanding in terms of memory and processing power. Therefore, we make a reduced version of this LM, pruning it with a probability threshold of 10^{-7}. The pruned LM is much smaller in size than the original, but this comes at the price of a higher perplexity, which rises from 286.86 to 413.62. Due to its smaller size, it can easily be used to generate word lattices on the evaluation data, which are rescored afterwards using the full unpruned LM. To demonstrate the extent to which they may affect the WER in practice, we perform an ASR evaluation on the IWSLT dev set using the pruned LM, before and after rescoring with the unpruned LM. The acoustic model in this experiment is the optimal model as established in section 4.2. The pruned LM yields in this evaluation a WER of 37.02%, a slight improvement over 39.95%, obtained in section 4.2, with a different LM. Rescoring with the full LM brings the WER further down to 33.69%.

5.2. Recurrent Neural Net Language Models

From a concatenation of all the whittled down text corpora of section 5.1, we train a Recurrent Neural Net Language Model, using the RNNLM toolkit [7]. Due to computational limitations, the vocabulary size for this model is reduced to 50000. The number of nodes in the hidden layer is set to 30. From the final rescored word lattices in section 5.1, N-best lists are generated, with N=100. For each of these 100 recognition hypotheses, the RNNLM is used to calculate a LM score S_{RNNLM}, which is interpolated with the original 4-gram LM score, resulting in the modified score S'.

$$S' = (1-\alpha) \cdot S_{ngram} + \alpha \cdot S_{RNNLM} \qquad (1)$$

This modified score is used to re-rank the N-best list, often changing which hypothesis is considered as the 'best'. The

interpolation factor α was optimised on the dev set, yielding a value of 0.25. Applying this RNNLM rescoring on the word lattices of section 5.1, yields an improvement in WER from 33.69% to 33.17%.

6. ASR System Setup

At this point, we have all the resources to build a finalised system: a large set of transcribed speech for acoustic model training, determined in section 2.2, and a large LM, optimised as described in section 5. The lay-out of our system is depicted in figure 3. All experiments performed with this

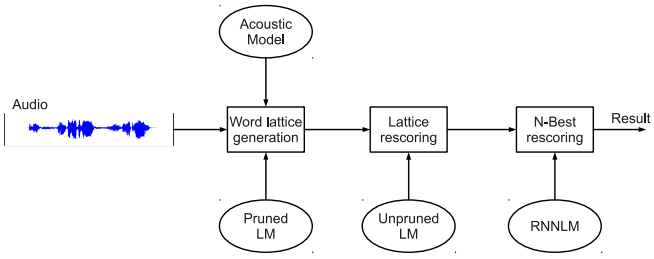


Figure 3: *A schematic overview of the adopted system.*

system, including the evaluations above and those that follow, have been performed using the KALDI Speech Recognition Toolkit [17]. For acoustic modelling, we first train up a GMM-HMM with 3000 context dependent states and 48000 Gaussians, using Speaker Adaptive Training (SAT), where fMLLR is used as the adaptation technique. In principle, it would be possible to assign multiple speeches to a single speaker, since the speaker's identity is given on the Europarl website. This only applies, however, to directly recorded speeches, i.e. untranslated ones. When the speaker is an interpreter, there is no trivial way to ascertain his/her identity. Therefore, we have made the simplifying assumption that each speech in the training data comes from a unique speaker. A feed-forward deep neural network is then trained in a DNN-HMM hybryd configuration, similar to the one used in [18]. This DNN has 6 hidden layers, each containing 2048 nodes. The softmax output layer of this network produces posterior probabilities over the 3000 context-dependent states of the HMM. The input at each time t consists of a stacking of the features in the context window $[t-5, t-4, \ldots t, \ldots t+4, t+5]$. Except for the addition of speaker adaptation, the features in each frame are produced as explained in section 4. Since the IWSLT test set is provided without segmentation into utterances, one can either generate a segmentation automatically, or perform recognition on entire TED-talks without segmentation. For the automatic segmentation, we use a voice activity detection system trained on 70 hours of English conversational speech from the AMI Meetings Corpus [19]. Speech and silence frames are modelled with diagonal covariance GMMs. A minimum duration constraint of 50ms is applied to each segment. For the segmentationless recognition, we use the same technique

	dev2012	tst2013	tst2013\E06
manual segment	27.02	35.27	29.18
auto segment	✗	39.28	33.58
no segment	28.16	36.21	30.24

Table 4: *The resulting WER's in % for several different evaluation sets, both when they are manually segmented, automatically segmented, or recognised in full (not segmented).*

as in [5], where we split an entire talk into overlapping segments, perform ASR on them, and dynamically merge the results into a single long recognition. In this case, segments are 40 seconds long and have an overlap of 20 seconds with each other. The results are listed in table 4. For the development set, no automatic segmentation was performed, since the manual segmentation was available for the official evaluation. There is one talk in the IWSLT test set, namely "E06_Nach-und-doch-so-Fern-Thomas-Mo", that is of very low quality. It has been recorded with a far-range microphone across a reverberant room, and contains quite a bit of non-speaker noise, e.g. coughing, rustling of paper and clothing, etc. Our system has not been designed to deal with such conditions, nor has it been tuned to them in any way, since the development set does not contain similar recordings. We therefore argue that this file unfairly skews the average test results. In tabel 4, the column "tst2013\E06" lists the results when this file is excluded from the evaluation. These error rates are more in line with those obtained on the dev set. The results in this table suggest that for TED talks, in the absence of a manual segmentation, a recognition performed on the whole talk is preferrable to using an automatically generated segmentation. We suspect, however, that this conclusion is fairly domain-specific. An automatic segmentation is essential for files with more music, jingles, applause, laughter, and other non-speaker noise.

7. Conclusion

We have presented the various components in the German ASR system, how they were set up, trained, and combined, to obtain accurate recognitions on the various data sets of the IWSLT evaluation task. Worthy of note is the acoustic model training, which was done almost entirely on publicly available data, without expert human transcriptions, using a lightly supervised training technique. Final evaluation on the unsegmented test set was performed in two different ways. Once with an automatically generated segmentation, and once without segmentation at all. It was found that, even though an oracle segmentation leads to optimal recognition results, avoiding segmentation altogether is preferrable to using an automatically generated one, when an oracle segmentation is not available.

8. References

[1] N. Braunschweiler, M. Gales, and S. Buchholz, "Lightly supervised recognition for automatic alignment of large coherent speech recordings," in *Proc. Interspeech*, September 2010, pp. 2222–2225.

[2] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech and Language*, vol. 16, no. 1, pp. 115–129, 2002.

[3] P. Placeway and J. Lafferty, "Cheating with imperfect transcripts," in *Proc. ICSLP*, vol. 4, 1996, pp. 2115–2118.

[4] P. Moreno and C. Alberti, "A factor automaton approach for the forced alignment of long speech recordings," in *Proc. ICASSP 2009.*, 2009, pp. 4869–4872.

[5] J. Driesen and S. Renals, "Lightly supervised automatic subtitling of weather forecasts," in *Proc. Automatic Speech Recognition and Understanding Workshop*, Olomouc, Czech Republic, December 2013.

[6] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proc. ACL*, Uppsala, Sweden, July 2010.

[7] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, Makuhari, Japan, September 2010.

[8] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, 1998.

[9] Y. Huang, D. Yu, Y. Gong, and C. Liu, "Semi-supervised GMM and DNN acoustic model training with multi-system combination and confidence re-calibration," in *Proc. Interspeech*, Lyon, France, 2013.

[10] T. Schultz, "GlobalPhone: A multilingual speech and text database developed at karlsruhe university," in *Proc. Interspeech*, Denver, Colorado, USA, 2002.

[11] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *Proc. ICASSP*, Vancouver, Canada, May 2013.

[12] "The website of the european parliament." [Online]. Available: http://europarl.europa.eu

[13] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, May 2008.

[14] A. Katsamanis, M. Black, P. Georgiou, L. Goldstein, and S. Narayanan, "Sailalign: Robust long speech-text alignment," in *Proc. of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, January 2011.

[15] "Iterative language model estimation: Efficient data structure & algorithms," in *Proc. Interspeech*, Brisbane, Australia, September 2008.

[16] S. F. Chen, , and J. Goodman, "An empirical study of smoothing techniques for language modeling." in *Proc. ACL*, Santa Cruz, USA, June 1996.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *Proc. ASRU*, Big Island, Hawaii, US, December 2011.

[18] P. Bell, P. Swietojanski, and S. Renals, "Multi-level adaptive networks in tandem and hybrid ASR systems," in *Proc. ICASSP*, Vancouver, Canada, May 2013.

[19] T. Hain, L. Burget, J. Dines, P. Garner, F. Grezl, A. Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "Transcribing meetings with the AMIDA systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 486–498, 2012.

NTT-NAIST SMT Systems for IWSLT 2013

Katsuhito Sudoh[*], *Graham Neubig*[†], *Kevin Duh*[†], *Hajime Tsukada*[*]

[*] NTT Communication Science Laboratories, Kyoto, Japan
[†] Nara Institute of Science and Technology, Nara, Japan
sudoh.katsuhito@lab.ntt.co.jp

Abstract

This paper presents NTT-NAIST SMT systems for English-German and German-English MT tasks of the IWSLT 2013 evaluation campaign. The systems are based on generalized minimum Bayes risk system combination of three SMT systems: forest-to-string, hierarchical phrase-based, phrase-based with pre-ordering. Individual SMT systems include data selection for domain adaptation, rescoring using recurrent neural net language models, interpolated language models, and compound word splitting (only for German-English).

1. Introduction

Spoken language is a very important and also challenging target for machine translation. MT tasks in the IWSLT evaluation campaign [1] focus on translating subtitles of speech from TED Talks. These subtitles are clean transcriptions without disfluencies that sometimes appeared in original talks. These talks can be expected to be similar to written texts that have been tackled in recent machine translation studies, as the talks are logically and syntactically well-organized compared to conversational speeches.

In our system this year, we focused on applying syntax-oriented translation technologies for statistical machine translation (SMT) such as forest-to-string translation and syntax-based pre-ordering. We also made several improvements to the base SMT models: domain adaptation by training data selection among different data sources; rescoring using recurrent neural network language models (RNNLMs); n-gram language model interpolation; compound word splitting for German compounds; and system combination of different types of SMT systems based on generalized minimum Bayes risk (GMBR) framework. This paper presents details of our systems and reports the results in German-English and English-German MT tasks in the evaluation campaign.

2. Translation Methods

The main feature of our system for this evaluation is that we perform translation using three different translation models and combine the results through system combination. Each of the three methods is described briefly below.

2.1. Phrase-based Machine Translation

Phrase-based machine translation (PBMT; [2]) models the translation process by splitting the source sentence into phrases, translating the phrases into target phrases, and reordering the phrases into the target language order. PBMT is currently the most widely used method in SMT as it is robust, does not require the availability of linguistic analysis tools, and achieves high accuracy, particularly for languages with similar syntactic structure.

2.2. Hierarchical Phrase-based Machine Translation

Hierarchical phrase-based machine translation (Hiero; [3]) expands the class of translation rules that can be used in phrase-based machine translation by further allowing rules with gaps that can be filled in a hierarchical fashion. Hiero is generally considered to be more accurate than PBMT on language pairs that are less monotonic, but also requires a significantly larger amount of memory and decoding time. As the German-English pair has a significant amount of reordering, particularly with movement of verbs, we can expect that Hiero will be able to handle these reorderings more appropriately in some cases.

2.3. Forest-to-string Machine Translation

Tree-to-string machine translation (T2S; [4]) performs translation by first syntactically parsing the source sentence, then translating from sub-structures of the parse to a string in the target language. Forest-to-string machine translation (F2S; [5]) generalizes this framework, making it possible to not only translate the single one-best syntactic parse, but a packed forest that encodes many possible parses, helping to pass along some of the ambiguity of parsing to be resolved during translation. While there are a number of proposed methods for incorporating source-side syntax into the translation process, here we use a method based on tree-to-string transducers [6].

Syntax-driven methods such as T2S and F2S are particularly useful for language pairs with extremely large amounts of reordering, as the syntactic parse can help guide the accurate re-ordering of entire phrases or clauses. On the other hand, these methods are highly dependent on parsing accuracy, and also have limits on the rules that can be extracted,

and are somewhat less robust than the previous two methods.

3. SMT Technologies

3.1. Training data selection

The target TED domain is different in both style and vocabulary from many of the other bitexts, e.g. Europarl, CommonCrawl (which we collectively call "general-domain" data[1]). To address this domain adaption problem, we performed adaptation training data selection using the method of [7].[2] The intuition is to select general-domain sentences that are similar to in-domain text, while being dis-similar to the average general-domain text.

To do so, one defines the score of an general-domain sentence pair (e, f) as [8]:

$$[IN_E(e) - GEN_E(e)] + [IN_F(f) - GEN_F(f)] \quad (1)$$

where $IN_E(e)$ is the *length-normalized* cross-entropy of e on the English in-domain LM. $GEN_E(e)$ is the length-normalized cross-entropy of e on the English general-domain LM, which is built from a sub-sample of the general-domain text. By taking a sub-sample (same size as the target-domain data), we reduce training time and avoid training and testing language models on the same general-domain data. Similarly, $IN_F(f)$ and $GEN_F(f)$ are the cross-entropies of f on Foreign-side LM. Finally, sentence pairs are ranked according to Eq. 1 and those with scores lower than some empirically-chosen threshold e.g. we choose this threshold by comparing BLEU on the dev set) are added together with the in-domain bitext for translation model training. Here, the LMs are Recurrent Neural Network Language Models (RNNLMs), which have been shown to outperform n-gram LMs in this problem [7].

3.2. Syntactic Rule-based Pre-ordering

Preordering is a method that attempts to first re-order the source sentence into a word order that is closer to the target. As German and English have significantly different word order, we can imagine that this will help our accuracy for this language pair.

3.2.1. German-to-English

We applied the clause restructuring method of Collins et al. [9] for German pre-ordering. The method is mainly based on moving German verbs in the end of clause structures towards the beginning of the clause. We re-implemented the method for German parse trees created using the Berkeley parser trained on TIGER corpus. We ignored some additional syntactic information such as subject markers and heads implemented in the original method of [9], because we used a different syntactic parser that did not provide this information.

3.2.2. English-to-German

We also tried to apply pre-ordering to English-to-German. We essentially did this by reversing the Collins German-to-English rules by moving some words towards the end of their siblings based on their part-of-speech tags as follows:

- in main clauses, `VB` words were moved,
- in subordinate clauses, `MD`, `VBP`, `VBD`, `VBZ` words were moved.

3.3. RNNLM Rescoring

Continuous-space language models using neural networks have attracted recent attention as a method to improve the fluency of output of MT or speech recognition. In our system, we used the recurrent neural network language model (RNNLM) of [10].[3] This model uses a continuous space representation over the language model state that is remembered throughout the entire sentence, and thus has the potential to ensure the global coherence of the sentence to the greater extent than simpler n-gram language models.

We incorporate the RNNLM probabilities through rescoring. For each system, we first output a 10,000-best list, then calculate the RNNLM log probabilities and add them as an additional feature to each translation hypothesis. We then re-run a single MERT optimization to find ideal weights for this new feature, and then extract the 1-best result from the 10,000-best list for the test set according to these new weights. The parameters for RNNLM training are tuned on the dev set to maximize perplexity, resulting in 300 hidden layers, 300 classes, and 4 steps of back-propogation through time.

3.4. German compound word splitting

German compound words present sparsity challenges for machine translation. To address this, we split German words following the general approach of [11]. The idea is to split a word if the geometric average of its subword frequencies is larger than whole word frequency. In our implementation, for each word, we searched for all possible decompositions into two sub-words, considering the possibility of deleting common German fillers "e", "es", and "s" (as in "Arbeit+s+tier"). For simplicity, we did not experiment with splitting into three or more sub-words as done in the `compound-splitter.perl` script distributed with the Moses package. The unigram frequencies for the subwords and whole word is computed from the German part of the bitext. This simple algorithm is especially useful for handling out-of-vocabulary and rare compound words that have high frequency sub-words in the training data. For the F2S sys-

[1] To give a sense of the domain difference, a 4-gram LM trained with Kneser-Ney smoothing on TED data gives a perplexity of 355 on the general domain data, compared to a perplexity of 99 on held-out TED data.

[2] Code/scripts available at http://cl.naist.jp/∼kevinduh/a/acl2013

[3] http://www.fit.vutbr.cz/~imikolov/rnnlm/

tem, sub-words are given the same POS tag as the original whole word.

In the evaluation campaign, we performed compound splitting only in the German-to-English task. We do not attempt to split German words for the English-to-German task, since it is non-trivial to handle recombination of German split words after reordering and translation.

3.5. GMBR system combination

We used a system combination method based on Generalized Minimum Bayes Risk optimization [12], which has been successfully applied to different types of SMT systems for patent translation [13]. Note that our system combination only picks one hypothesis from an N-best list and does not generate a new hypothesis by mixing partial hypotheses among the N-best.

3.5.1. Theory

Minimum Bayes Risk (MBR) is a decision rule to choose hypotheses that minimize the expected loss. In the task of SMT from a French sentence (f) to an English sentence (e), the MBR decision rule on $\delta(f) \rightarrow e'$ with the loss function L over the possible space of sentence pairs ($p(e, f)$) is denoted as:

$$\underset{\delta(f)}{\operatorname{argmin}} \sum_{e} L(\delta(f)|e)p(e|f) \tag{2}$$

In practice, we approximate this using N-best list $N(f)$ for the input f.

$$\underset{e' \in N(f)}{\operatorname{argmin}} \sum_{e \in N(f)} L(e'|e)p(e|f) \tag{3}$$

Although MBR works effectively for re-ranking single system hypotheses, it is challenging for system combination because the estimated $p(e|f)$ from different systems cannot be reliably compared. One practical solution is to use uniform $p(e|f)$ but this does not achieve Bayes Risk. GMBR corrects by parameterizing the loss function as a linear combination of sub-components using parameter $\boldsymbol{\theta}$:

$$L(e'|e;\boldsymbol{\theta}) = \sum_{k=1}^{K} \theta_k L_k(e'|e) \tag{4}$$

For example, suppose the desired loss function is "1.0−BLEU". Then the sub-components could be "1.0−precision(n-gram) $(1 \leq n \leq 4)$" and "brevity penalty".

Assuming uniform $p(e|f)$, the MBR decision rule can be denoted as:

$$\begin{aligned} &\underset{e' \in N(f)}{\operatorname{argmin}} \sum_{e \in N(f)} L(e'|e;\boldsymbol{\theta}) \frac{1}{|N(f)|} \\ &= \underset{e' \in N(f)}{\operatorname{argmin}} \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e'|e) \end{aligned} \tag{5}$$

To ensure that the uniform hypotheses space gives the same decision as the original loss in the true space $p(e|f)$, we use a small development set to tune the parameter $\boldsymbol{\theta}$ as follows. For any two hypotheses e_1, e_2, and a reference translation e_r (possibly not in $N(f)$) we first compute the true loss: $L(e_1|e_r)$ and $L(e_2|e_r)$. If $L(e_1|e_r) < L(e_2|e_r)$, then we would want $\boldsymbol{\theta}$ such that:

$$\sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_1|e) < \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_2|e) \tag{6}$$

so that GMBR would select the hypothesis achieving lower loss. Conversely if e_2 is a better hypothesis, then we want opposite relation:

$$\sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_1|e) > \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_2|e) \tag{7}$$

Thus, we directly compute the true loss using a development set and ensure that our GMBR decision rule minimizes this loss.

3.5.2. Implementation

We implement GMBR for SMT system combination as follows.

First we run SMT decoders to obtain N-best lists for all sentences in the development set, and extract all pairs of hypotheses where a difference exists in the true loss. Then we optimize $\boldsymbol{\theta}$ in a formulation similar to a Ranking SVM [14]. The pair-wise nature of Eqs. 6 and 7 makes the problem amendable to solutions in "learning to rank" literature [15]. We used BLEU as the objective function and the sub-components of BLEU as features (system identity feature was not used). There is one regularization hyperparameter for the Ranking SVM, which we set by cross-validation over the development set (`dev2010`).

3.6. What Didn't Work Immediately

We also tried several other methods that did not have a clear positive effect and were thus omitted from the final system. For example, we attempted to improve alignment accuracy using the discriminative alignment method proposed by [16] training on the 300 hand-aligned sentences.[4] However, while this provided small gains in alignment accuracy on a held-out set, the gains were likely not enough, and MT results were inconclusive. We also attempted to use the reordering method of [17] as implemented in lader,[5] again trained on the same 300 hand-aligned sentences, but increases in reordering accuracy on a held-out set were minimal. We believe that both of these techniques are promising, but require a larger set of hand-aligned data to provide gains large enough to appear in MT results.

[4] `http://user.phil-fak.uni-duesseldorf.de/~tosch/downloads.html`

[5] `http://phontron.com/lader`

4. Experiments

4.1. Setup

4.1.1. System overview

We used three individual SMT systems for each language pairs: forest-to-string (F2S), hierarchical phrase-based (Hiero), and phrase-based with pre-ordering (Preorder). In some of our comparisons we also use simple phrase-based translation without preordering (PBMT). F2S was implemented with Travatar [18] and Preorder, PBMT, and Hiero were implemented using Moses [19].

For the Moses models, we generally used the default settings, but with Good-Turing phrase table smoothing. For F2S translation we used Egret[6] as a parser, and created forests using dynamic pruning including all edges that occurred in the 100-best hypotheses. We trained the parsing model using the Berkeley parser over the Wall Street Journal section of the Penn Treebank[7] for English, and TIGER corpus [20] for German. For model training, the default settings for Travatar were used, with the exception of changing the number of composed rules to 6 and using Kneser-Ney rule table smoothing.

All systems were evaluated using the standard BLEU score [21] and also RIBES [22], a metric designed specifically to show whether reordering is being performed properly. All systems were optimized towards BLEU score. We measure statistical significance between results with bootstrap resampling with $p > 0.05$. Bold numbers in each table indicate the best system, and all systems that do not show a statistically significant difference from the best system [23].

All words were lowercased prior to translation, and finally recased by a SMT-based recaser as implemented in Moses.

4.1.2. Translation models

We trained the translation models using WIT^3 training data (138,499 sentences) and 1,000,000 sentences selected over other bitexts (Europarl, News Commentary, and Common Crawl) by the method described in 3.1.

4.1.3. Language models

We used two types of word n-gram language models of German and English: interpolated 6-gram and Google 5-gram.

The interpolated 6-gram LMs were from linear interpolation of several 6-gram LMs on different data sources (WIT^3, Europarl, News Commentary, Common Crawl, Common News, and MultiUN). The interpolation weights were optimized for test set perplexities on the development set, using `interpolate-lm.perl` in Moses. Individual 6-gram LMs were trained by SRILM with modified Kneser-Ney smoothing.

[6] `https://github.com/neubig/egret/`
[7] `http://www.cis.upenn.edu/~treebank/`

System	tst2011	tst2012	tst2013
Combination	26.04	22.86	24.60
F2S	26.27	22.59	24.34
Hiero	24.55	20.66	22.80
Preorder	25.30	21.84	24.08

Table 1: Official BLEU results for English-to-German (case-sensitive).

The Google 5-gram LMs were from Google Web 1T N-grams. We limited vocabulary words to those with 8,192 or more in unigram counts and all words were mapped to lowercase. Then we trained 5-gram LMs with Witten-Bell smoothing.

4.1.4. Recaser models

The Moses-based recaser model for both English and German were trained by `train-recaser.perl` using monolingual resources (WIT^3, Europarl, News Commentary, Common Crawl, Common News, and MultiUN).

4.2. Full System Results

Our full system was the combination of F2S, Hiero, and Preorder. Tables 1 and 2 show the evaluation results for the official test sets in German-to-English and English-to-German, respectively. In German-to-English, each individual system showed similar performance in BLEU and the system combination achieved much higher BLEU score, 2.8 points higher than Preorder. In English-to-German, F2S showed the best performance among the three individual systems and the system combination was not so effective as in German-to-English.

The contributions of individual systems can be measured by the number of each system's output chosen by the system combination, as shown in Table 3. These results suggest:

- When one system is much better than the others, our system combination highly relies on the best system and has a little room for improvement. (English-to-German)

- When the individual systems are different each other, the voting-like effect of our system combination improves the overall performance even if individual performances are similar. (German-to-English)

These findings are similar to our system combination results in English-Japanese translation [24].

With respect to recasing, slight BLEU drops were found between case-sensitive and case-insensitive evaluation as shown in Table 4. There was a larger drop in English-German than German-English, due to the large number of required recasing for German nouns.

System	tst2013
Combination	25.83
F2S	23.03
Hiero	22.76
Preorder	23.04

Table 2: Official BLEU results for German-to-English (case-sensitive, without disfluency).

Task	F2S	Hiero	Preorder	ALL
English-German	868	0	125	993
German-English	304	142	916	1,362

Table 3: Number of each system's outputs chosen by system combination for tst2013.

	En-De	De-En
case-sensitive	24.60	25.83
case-insensitive	25.79	26.45

Table 4: Official BLEU results by Combination systems on tst2013 set with case-sensitive and case-insensitive evaluation (without disfluency).

4.3. Effect of Data Selection

Experimental results on adaptation training data selection is shown in Table 5. By adding 1 million (1M) general-domain sentences, we improve a baseline de-en PBMT system (which is only trained from in-domain TED data) from 27.26 to 28.09 BLEU. We improve from 21.53 to 22.11 BLEU in the en-de PBMT system. This 1M general-domain data is combined with the in-domain TED bitext in subsequent system building, which required sufficiently fewer computational resources than using the entire general-domain data (especially for the F2S system).

Interestingly, we have found the improvements in Table 5 are not as large as that reported in [7] despite the similar task setup. The results are not directly comparable due to different dev/test splits and random initializations. Nevertheless, it has come to our attention that the random sampling of general-domain data for $GEN_E(e)$ and $GEN_F(f)$ in Eq. 1 appears to cause large differences in the subsequent RNNLMs. This is because the RNNLMs are highly optimized on perplexity. We suspect that using only $IN_E(e)$ and $IN_F(f)$ as the sentence selection criteria (or using the simpler n-grams for $GEN_E(e)$ and $GEN_F(f)$ values) may give more stable results, though we have not tried comprehensive experiments to validate this.

4.4. Translation Method Comparison

In this section, we provide a brief comparison of the three translation methods mentioned in Section 2 on tst2010 data. For all systems we used the TED data and 1M selected sentences for training, and used the language model described in the previous section. None of the results include RNNLM, and are somewhat preliminary, so they do not match our final submission exactly.

	Number of Selected General-domain Sentences					
	0	100k	500k	1M	2M	all
de-en	27.26	27.51	27.55	28.09	27.43	27.44
en-de	21.53	21.58	21.73	22.11	21.92	22.09

Table 5: BLEU results for adaptation training data selection. These are tst2010 results using a preliminary PBMT system, so they are not directly comparable to other results in this paper.

	en-de		de-en	
	BLEU	RIBES	BLEU	RIBES
PBMT	23.11	80.56	**30.51**	**84.68**
Hiero	23.33	**81.17**	**30.54**	**84.51**
F2S	**24.30**	**81.09**	**30.37**	83.44

Table 6: A comparison between different translation methods with exactly matched training conditions.

	Baseline	+Splitting
PBMT	30.36	**30.51**
Hiero	30.22	**30.54**
F2S	29.82	**30.36**

Table 7: BLEU results for compound splitting.

The results are shown in Table 6. From these results, we can see that given exactly the same data, alignments, and language model, F2S achieved the highest accuracy on English-German, and PBMT and Hiero achieved higher accuracy on German-English. For English-German, we noticed that the F2S system did a significantly better job of accurately generating verbs at the end of the German sentence, demonstrating its superior capability for reordering. For German-English, on the other hand, F2S achieved a somewhat counter-intuitive low score on the reordering-based measure RIBES. Upon an analysis of the results, we found that the F2S system was largely getting the reordering right, but occasionally making big changes in reordering large clauses that were not reflected in the German reference. It is likely that if we optimized towards RIBES, or a combination of BLEU and RIBES [25] we might get better results.

4.5. Effect of Compound Splitting

Next, we examine the effect of compound splitting for German-English translation. From the results in Table 7, we can see that compound splitting provides a gain for all systems, and particularly so for F2S translation.

	en-de		de-en	
	n-gram	+RNNLM	n-gram	+RNNLM
PBMT	23.11	**23.81**	30.51	**31.03**
Hiero	23.33	**24.31**	30.54	**31.80**
F2S	24.30	**25.02**	30.48	**30.85**

Table 8: BLEU results for RNNLM rescoring.

4.6. Effect of RNNLM

Next, we examine the effect of adding RNNLM to the translation accuracy. From the results in Table 8, we can see that the RNNLM provided significant gains in all cases, ranging from 0.4-1.3 BLEU points. Examining the results manually, we it was difficult to identify one clear reason for the improvements in the scores, but we did see some subjective improvements in agreement between prepositions in coordinate structures, and less collapse of syntactic structure around unknown words.

5. Conclusion

We used various SMT technologies for this year's evaluation campaign. Most of them had positive effects on the final translation performance. The forest-to-string SMT had the largest contribution in English-to-German, and the GMBR system combination largely increased the performance in German-to-English.

6. References

[1] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 10th IWSLT Evaluation Campaign," in *Proc. IWSLT 2013*, 2013.

[2] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proc. HLT*, Edmonton, Canada, 2003, pp. 48–54.

[3] D. Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, 2007.

[4] Y. Liu, Q. Liu, and S. Lin, "Tree-to-string alignment template for statistical machine translation," in *Proc. ACL*, 2006.

[5] H. Mi and L. Huang, "Forest-based translation rule extraction," in *Proc. EMNLP*, 2008, pp. 206–214.

[6] J. Graehl and K. Knight, "Training tree transducers," in *Proc. HLT*, 2004, pp. 105–112.

[7] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation data selection using neural language models: Experiments in machine translation," in *Proc. ACL*, 2013.

[8] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proc. EMNLP*, 2011, pp. 355–362.

[9] M. Collins, P. Koehn, and I. Kucerova, "Clause restructuring for statistical machine translation," in *Proc. ACL*, 2005.

[10] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Proc. 11th InterSpeech*, 2010, pp. 1045–1048.

[11] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proc. EACL*, 2003.

[12] K. Duh, K. Sudoh, X. Wu, H. Tsukada, and M. Nagata, "Generalized minimum bayes risk system combination," in *Proc. IJCNLP*, 2011, pp. 1356–1360.

[13] K. Sudoh, K. Duh, H. Tsukada, M. Nagata, X. Wu, T. Matsuzaki, and J. Tsujii, "NTT-UT statistical machine translation in NTCIR-9 PatentMT," in *Proc. NTCIR*, 2011.

[14] T. Joachims, "Training linear svms in linear time," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 217–226.

[15] C. He, C. Wang, Y.-X. Zhong, and R.-F. Li, "A survey on learning to rank," in *Machine Learning and Cybernetics, 2008 International Conference on*, vol. 3. IEEE, 2008, pp. 1734–1739.

[16] J. Riesa and D. Marcu, "Hierarchical search for word alignment," in *Proc. ACL*, 2010, pp. 157–166.

[17] G. Neubig, T. Watanabe, and S. Mori, "Inducing a discriminative parser to optimize machine translation reordering," in *Proc. EMNLP*, Korea, July 2012, pp. 843–853.

[18] G. Neubig, "Travatar: A forest-to-string machine translation engine based on tree transducers," in *Proc. ACL Demo Track*, Sofia, Bulgaria, August 2013.

[19] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proc. ACL*, Prague, Czech Republic, 2007, pp. 177–180.

[20] S. Brants, S. Dipper, P. Eisenberg, S. Hansen-Schirra, E. König, W. Lezius, C. Rohrer, G. Smith, and H. Uszkoreit, "Tiger: Linguistic interpretation of a german corpus," *Research on Language and Computation*, vol. 2, no. 4, pp. 597–620, 2004.

[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. ACL*, Philadelphia, USA, 2002, pp. 311–318.

[22] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada, "Automatic evaluation of translation quality for distant language pairs," in *Proc. EMNLP*, 2010, pp. 944–952.

[23] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proc. EMNLP*, 2004.

[24] K. Sudoh, H. Tsukada, M. Nagata, S. Hoshino, and Y. Miyao, "NTT-NII statistical machine translation in NTCIR-10 PatentMT," in *Proc. NTCIR*, 2013.

[25] K. Duh, K. Sudoh, X. Wu, H. Tsukada, and M. Nagata, "Learning to translate with multiple objectives," in *Proc. ACL*, 2012.

The 2013 KIT IWSLT Speech-to-Text Systems for German and English

Kevin Kilgour, Christian Mohr, Michael Heck, Quoc Bao Nguyen, Van Huy Nguyen, Evgeniy Shin, Igor Tseyzer, Jonas Gehring, Markus Müller, Matthias Sperber, Sebastian Stüker and Alex Waibel

Institute for Anthropomatics
Karlsruhe Institute of Technology
Karlsruhe, Germany

{kevin.kilgour|christian.mohr|heck|quoc.nguyen|van.nguyen|eugene.sheen}@kit.edu
{igor.tseyzer|jonas.gehring|m.mueller|matthias.sperber|sebastian.stueker|waibel}@kit.edu

Abstract

This paper describes our English *Speech-to-Text* (STT) systems for the 2013 IWSLT TED ASR track. The systems consist of multiple subsystems that are combinations of different front-ends, e.g. MVDR-MFCC based and lMel based ones, GMM and NN acoustic models and different phone sets. The outputs of the subsystems are combined via confusion network combination. Decoding is done in two stages, where the systems of the second stage are adapted in an unsupervised manner on the combination of the first stage outputs using VTLN, MLLR, and cMLLR.

Index Terms: speech recognition, IWSLT, TED talks, evaluation system, system development

1. Introduction

[1] The *International Workshop on Spoken Language Translation* (IWSLT) offers a comprehensive evaluation campaign on spoken language translation. One part of the campaign focuses on the translation of TED Talks (http://www.ted.com/talks), short 5-25min presentations by people from various fields related in some way to Technology, Entertainment, and Design (TED) [2]. In order to evaluate different aspects of this task IWSLT organizes several evaluation tracks on this data covering the aspects of automatic speech recognition (ASR), machine translation (MT), and the full-fledged combination of the two of them into speech translation systems.

The goal of the TED ASR track is the automatic transcription of TED lectures on a given segmentation, in order to interface with the machine translation components in the speech-translation track. The quality of the resulting transcriptions are measured in word error rate (WER).

In this paper we describe our English ASR systems with which we participated in the TED ASR track of the 2013 IWSLT evaluation campaign. This year, our system is a further development of our last year's evaluation system [3] and makes use of system combination and cross-adaptation, by utilising both GMM and Neural Network acoustic models which are trained with different acoustic front-ends and employ different phoneme sets. We also included TED talks available via TED's website by training on them in a slightly supervised manner.

We submitted primary systems for both the German and English evaluations.

The rest of this paper is structured as follows. Section 2 describes the data that our system was trained and tested on. This is followed by section 3 which provides a description of the two acoustic front-ends used in our system and section 4 which describes our segmentation setup. An overview of the techniques used to build our acoustic models is given in section 5. We describe the language model used for this evaluation in section 6 and our decoding strategy and results are presented in sections 7 and 8.

2. Data Resources

2.1. Training Data

For acoustic model training we used the following English data sources:

- 200 hours of Quaero training data from 2010 to 2012.
- 18 hours of various noise data, such as snippets of applause and music.
- 158 hours of data downloaded from the TED talks website that was released before the cut-off date of 31 December 2010, including the corresponding subtitles provided by the TED conferences archive.

and the following German data sources:

- 179 hours of Quaero training data from 2010 to 2012.
- 24 hours of broadcast news data

These training set or subsets hereof are also used for the training of the automatic segmenters, that are applied to the evaluation data before decoding.

For English language model training and vocabulary selection, we used the subtitles of TED talks and text data from

Text corpus	# Words
TED	3M
News + News commentary	2,114M
GIGA parallel	523M
Gigaword 4	1,800M
UN + Europarl	376M
Google Books Ngrams (subset)	(1000M ngrams)

Table 1: English language modeling data after cleaning and data selection. The total number of words was 4.8 billion, not counting Google Books.

Text corpus	# Words
TED (translated)	2,259k
Callhome	150k
Europarl	47,306k
HUB5	19k
MultiUN	5,849k
News+News Commentary	284,415k
ECI	12,652k
Euro Language Newspaper	86,785k
German Political Speeches	5,514k
Common Crawl	47,046k
Google Web Ngrams	1.3T

Table 2: German language modeling data after cleaning and data selection. In total, we used 492 million words, not counting Google Ngrams.

various sources (see Table 1) and for the German language model training and vocabulary selection, we used translated subtitles of TED talks and text data from various sources (see Table 2).

2.2. Test Data

Table 3 describes three test sets ("tst2011", "tst2012" and "tst2013") used for this year's English evaluation campaign, as well as our development set for system development and parameter optimization ("dev2012"). "tst2011" is comprised of TED talks newer than December 2010 and serves as progress test set to measure the improvement in systems from 2011 onwards. "tst2012" is last year's evaluation set, and "tst2013" is a collection of some of the most recent recordings made available by TED. All test sets were used with the original pre-segmentation provided by the IWSLT organizers, except for this year's evaluation set ("tst2013") which has been segmented automatically before decoding. For the German system on a single test set "dev2013" was available.

3. Feature Extraction

Our systems are built using several different front ends that use various inputs for computing deep bottle neck features.

Set	#talks	#utt	dur	dur/utt
dev2012	10	1144	1.7h	5.4s
tst2011	8	818	1.1h	4.9s
tst2012	11	1124	1.7h	5.6s
tst2013	28	1438	4.2h	10.5s

Table 3: Statistics of the development set ("dev2012") and the test sets ("tst2011", "tst2012" and "tst2013"), including the total number of talks (*#talks*), the total number of utterances (*#utt*), the overall speech duration (*dur*), and average speech duration per utterance (*dur/utt*). "tst2013" has been segmented automatically.

The two main input variants, each using a frame shift of 10ms and a frame size of 32ms, are the MFCC+MVDR (M2) features that have been shown to be very effective when used in BNFs [4] and standard lMEL features which generally outperform MFCCs as DBNF inputs. These standard features are often augmented by tonal features. In [**?**] we demonstrate, that the addition of tonal (T) features not only greatly reduces the WER on tonal languages like Vietnamese and Cantonese but also results in small gains on non-tonal languages like English.

13 frames (+-6 frames) are stacked as the DBNF input which consists of 4-5 hidden layers each containing 1200-1600 units followed by a 42 unit bottleneck, a further 1200-1600 unit hidden layer and an output layer of 6000 context dependent phone states for the German systems and 8000 for the English systems. The first 4-5 hidden layers are pretrained layer-wise as denoising autoencoders after which the network the finetuned as a whole [5]. As can be seen in figure 1 the layers after the bottlenet are discareded and 13 (+-6) bottleneck frames are stacked and reduced back down to a 42 dimensional input feature using LDA.

4. Automatic Segmentation

For this year's ASR track, the evaluation set was provided without manual sentence segmentation, thus automatic segmentation of the target data was mandatory. We evaluated the effectiveness of three different approaches to automatic segmentation of audio data, which are:

a) *Decoder based* segmentation on hypotheses. A fast decoding pass with one of our development systems was done to determine speech and non-speech regions as in [6]. Segmentation is performed by consecutively splitting segments at the longest non-speech region with a minimal duration of at least 0.3 seconds. *b*) *GMM based* segmentation using speech, non-speech and silence models. This method uses a Viterbi decoder and GMM models for the three aforementioned categories of sounds. The general framework is based on the one in [7], which was likewise derived from [8]. In contrast to the previous work, we made use of additional features such as a zero crossing rate. *c*) *SVM based* segmentation using speech and non-speech models, using the frame-

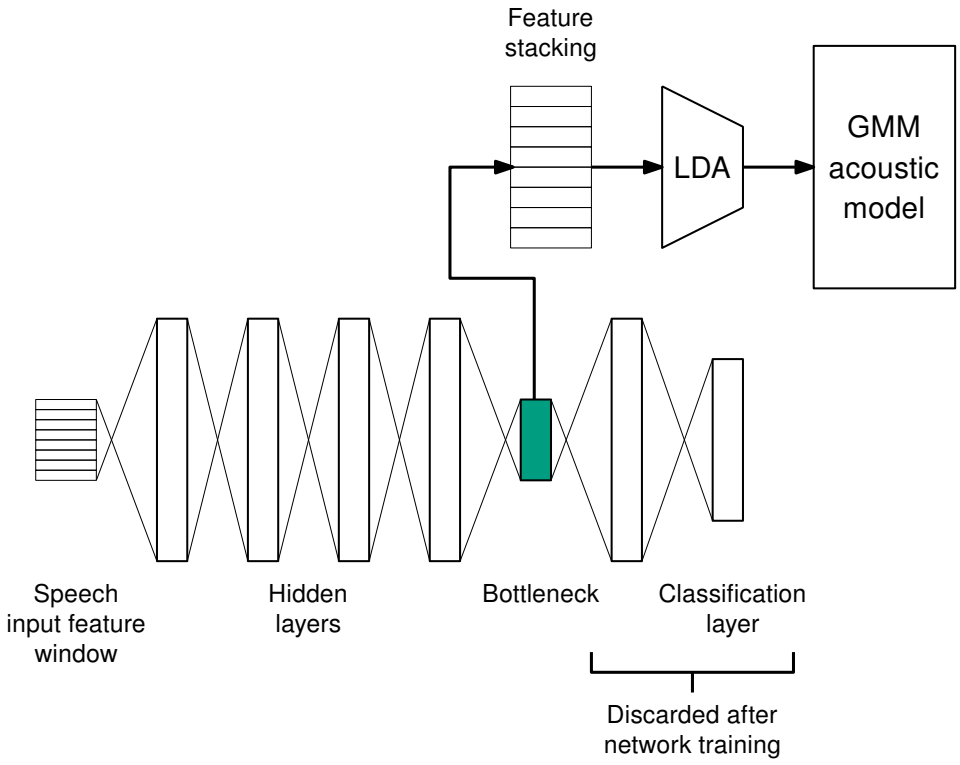


Figure 1: *Overview of our standard DBNF setup.*

work introduced in [7]. The pre-processing makes use of an LDA transformation on feature vectors after frame stacking to effectively incorporate temporal information. The SVM classifier is trained with the help of LIBSVM [9]. A 2-phased post-processing is applied for final segmentation generation.

Table 4 shows the decoding performance of a confusion network combination of hypotheses generated by five development systems after a first pass decoding on the "dev2012" set, for each preliminary application of the various techniques for segmentation.

Segmentation	WER	#utt	dur	dur/utt
Manual	13.2%	1144	1.71h	5.4s
Decoder based	13.8%	594	1.83h	11.1s
SVM based	13.9%	431	1.78h	14.9s
GMM based	14.3%	695	1.77h	9.2s

Table 4: Decoding performance on and statistics of the development set ("dev2012") after automatic segmentation, including the word error rate (*WER*), the total number of utterances (*#utt*), the overall speech duration (*dur*), and average speech duration per utterance (*dur/utt*).

On the English development set the decoder based approach resulted in the best performance in terms of WER, so we decided in favor of the latter for application on the evaluation set. For the German system we used the SVM based segmenters since it performed best on the German development set.

5. Acoustic Modeling

We trained several different acoustic models for each language.

5.1. Data Preprocessing

For the TED data only subtitles were available so the data had to be segmented prior to training. In order to split the data into sentence-like chunks, it was decoded to discriminate speech and non-speech and a forced alignment given the subtitles was done where only the relevant speech parts detected by the decoding were used. The procedure is the same that has been applied in [10].

5.2. AM training Setup

The models of all systems are context-dependent quinphones with three states per phoneme, using a left-to-right HMM topology without skip states. All English acoustic models initially use 8,000 distributions and codebooks derived from decision-tree based clustering of the states of all possible quinphones. The German acoustic models use 6000 distributions and codebooks.

The GMM models were trained by using incremental

splitting of Gaussians training (MAS) [11], followed by optimal feature space training (OFS) which is a variant of *semi-tied covariance* (STC) [12] training using one global transformation matrix, and finally refined by one iteration of Viterbi training. All models further use vocal tract length normalization (VTLN).

We trained multiple different GMM acoustic models by combining different front-ends and different phoneme sets. Section 7 elaborates the details of our system combination.

5.3. Hybrid Acoustic Model

We experimented with using neural network acoustic models. Using the same techniques described in the deep bottleneck layer section we trained neural networks on various input features and with different topologies. Our best setups used deep bottleneck features stacked over a window of 13 frames, with 4-5 1600-2000 unit hidden layers and an output layer containing 6016 context dependent phonestates. The deep bottleneck features were extracted using an MLP with 5 1600 unit hidden layers prior to the 42 unit bottleneck layer. Its input was 40 lMel (or MVDR+MFCC) and 14 tone features stacked over a 13 frame window. Both neural networks were pretrained as denoising autoencoders. On the eval2010 test set this system had a WER of 14.61%, which is 0.5% better than this best non hybrid single pass system.

5.4. Pronunciation Dictionary

We used two different phoneme sets. The first one is based on the CMU dictionary[1] and is the same phoneme set as the one used in last years system. It consists of 45 phonemes and allophones. The second phoneme set is derived from the BEEP dictionary[2] and contains 44 phonemes and allophones. Both sets use 7 noise tags and one silence tag each. For the CMU phoneme set we generated missing pronunciations with the help of FESTIVAL [13], while for the BEEP dictionary we used Sequitur [14] instead. Both grapheme to phoneme converters were trained on subsets of the respective dictionaries.

5.5. Grapheme System

We built grapheme-based recognizer for both English and German. In order to built the Englsih grapheme-based dictionary, we used a data-driven approach to cluster the most common combinations of letters in order to better reflect the specifics of the English language. These clusters contain for instance combinations such as sch, sh or th. We added these in addition to all the letters of the English alphabet to the set of phones.

Using this dictionary, be trained a system using flatstart training on the training data of the 2011 training set. After doing the context-independent flatstart training, we built a context-dependent system on top of that.

As our best result, we archived to get a WER of 31.8% using a clustertree with 6000 states. Since this WER is quite high compared to the WER of our other systems, we decided not to include this system either in our system-combination or the submission.

The German grapheme system on the other hand performed only slighty worse than our phoneme based system and resulted in overall gains when included in the final system combination.

5.6. BMMIE training

In order to improve the performance of acoustic model, the Boosted Maximum Mutual Information Estimation training (BMMIE) [15] is applied, it is a modified form of the Maximum Mutual Information (MMI) [16]. We wrote lattices for discriminative training using a small unigram language model as in [17]. After lattices generating, the BMMIE training is applied for three iterations with boosting factor b=0.5. This approach resulted in about 0.6% WER improvement for 1st-pass sytems and about 0.4% WER for 2nd-pass systems.

6. Lanuage Models and Search Vocabulary

Language modeling was performed by building separate language models for all (sub-)corpora using the SRILM toolkit [18] with modified Kneser-Ney smoothing. These were then linearly interpolated, with interpolation weights tuned using held-out data from the TED corpus.

6.1. Subword Language Model for German

In order to select a sub-word vocabulary we first perform compound splitting on all the text corpora and tag the split compounds. Linking morphemes are attached to the proceeding word. *Wirtschaftsdelegationsmitglieder* is, for example, split into *Wirtschafts+ Delegations+ Mitglieder (eng: members of the economic delegation)*.

Our compound splitting algorithm requires a set of valid sub-words and selects the best split from all possible splits by maximizing the sum of the squares of all sub-word lengths [19]. For the word *Konsumentenumfrage* this heuristic would correctly choose *Konsumenten Umfrag* over *Konsum Enten Umfrage*.

As a set of valid sub-words we selected the top k words from a ranked word-list generated in the same mannar as our English vocabulary. After applying coumpound splitting to all our text corpora the same maximum likelihood vocabulary selection method is used again to select the best vocabulary from this split corpora resulting in a ranked vocabulary containing both full words and sub-words tagged with a "+".

Pronunciations missing from the initial dictionary are created with both Festival and Mary [20]. The sub-word language model is trained on the split corpora and tuning text analogous to the English language model.

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[2] ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/dictionaries/beep.tar.gz

System	Dev2012	Eval2011	Eval2012
M2+T-CMU	15.9	11.6	11.7
lMEL+T-CMU	16.1	11.4	11.4
M2+T-DLabel-CMU	15.8	11.2	11.5
M2+T-BEEP	16.2	12.0	12.6
lMEL+T-BEEP	16.1	12.2	12.6
M2+T-hyb-CMU	16.5	11.9	11.6
M2+T-hyb-BEEP	16.9	12.4	12.4
CNC-BEEP-01	13.7	9.8	9.5
M2+T-CMU	14.7	10.3	10.3
lMEL+T-CMU	15.0	10.2	10.1
M2+T-DLabel-CMU	14.5	10.3	10.1
M2+T-BEEP	14.7	10.8	10.5
lMEL+T-BEEP	14.4	10.6	10.6
CNC-BEEP-02	13.3	9.3	9.2
ROVER	13.3	9.2	9.0

Table 5: Results for English language on development data and evaluation data.

System	Dev	Eval
M2-P-bmmie-i3	21.00	29.40
M2+T-P-bmmie-i4	20.80	30.80
M2+T-G-bmmie-i3	21.70	29.80
M2-hyb-P	21.40	30.50
lMEL+T-P-bmmie-i3	21.10	29.70
lMEL-hyb-P	20.20	29.20
M2-G-bmmie-vit	22.90	30.70
CNC-01	18.60	26.70
M2-P-bmmie-i3-SAT	19.90	27.90
M2+T-P-bmmie-i4-SAT	19.60	27.80
M2+T-G-bmmie-i3-SAT	20.50	27.90
lMEL+T-P-bmmie-i3-SAT	20.10	27.80
M2-G-bmmie-vit-SAT	21.70	29.00
CNC-02	18.30	26.40
ROVER	18.30	26.30

Table 6: Results for German language on development data und evaluation data.

7. Decoding Setup

The decoding was performed with the *Janus Recognition Tool-kit* (JRTk) developed at Karlsruhe Institute of Technology and Carnegie Mellon University [21]. Our decoding strategy is based on the principle of system combination and cross-system adaptation. System combination works on the principle that different systems commit different errors that cancel each other out. Cross-system adaptation profits from the fact that the unsupervised acoustic model adaptation works better when performed on output that was created with a different system that works approximately equally well [22]. The final step in our system decoding set-up is the ROVER combination of several outputs [23].

8. Results

We evaluated our systems on the IWSLT test sets 2011 (tst2011), 2012 (tst2012) and the 2012 dev set. We used the dev2012 set as development set and for parameter optimization and the eval 2012 set to compare our system with last years evaluation results (see table 5). Last year our best system had a WER of 12% on the eval 2012 set which we were able to reduce to 9% with this year's evaluation system.

9. Conclusions

In this paper we presented our English and German LVCSR systems, with which we participated in the 2013 IWSLT evaluation.

10. Acknowledgements

'*Research Group 3-01*' received financial support by the '*Concept for the Future*' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative. The work leading to these results has received funding from the European Union under grant agreement $n \circ 287658$. This work was partly realized within the Quaero Programme, funded by OSEO, French State agency for innovation.

11. References

[1] S. Stüker, K. Kilgour, and F. Kraft, "Quaero 2010 speech-to-text evaluation systems," in *High Performance Computing in Science and Engineering '11*, W. E. Nagel, D. B. Kröner, and M. M. Resch, Eds. Springer Berlin Heidelberg, 2012, pp. 607–618.

[2] S. Stüker, F. Kraft, C. Mohr, T. Herrmann, E. Cho, and A. Waibel, "The KIT lecture corpus for speech translation," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2012, to appear.

[3] Christian Saam, Christian Mohr, Kevin Kilgour, Michael Heck, Matthias Sperber, Keigo Kubo, Sebastian Stüker, Sakriani Sakti, Graham Neubig, Tomoki Toda, Satoshi Nakamura, and lex Waibel, "The 2012 KIT and KIT-NAIST English ASR Systems for the IWSLT Evaluation," in *International Workshop on Spoken Language Translation (IWSLT)*, Dec. 2012.

[4] K. Kilgour, I. Tseyzer, Q. B. Nguyen, and A. Waibel, "Warped minimum variance distortionless response based bottle neck features for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6990–6994.

[5] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-

encoders," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE*, 2013.

[6] S. Stüker, C. Fügen, F. Kraft, and M. Wölfel, "The ISL 2007 English Speech Transcription System for European Parliament Speeches," in *Proceedings of the 10th European Conference on Speech Communication and Technology (INTERSPEECH 2007)*, Antwerp, Belgium, August 2007, pp. 2609–2612.

[7] M. Heck, C. Mohr, S. Stker, M. Mller, K. Kilgour, J. Gehring, Q. Nguyen, V. Nguyen, and A. Waibel, "Segmentation of telephone speech based on speech and non-speech models," in *Speech and Computer*, ser. Lecture Notes in Computer Science, M. elezn, I. Habernal, and A. Ronzhin, Eds. Springer International Publishing, 2013, vol. 8113, pp. 286–293.

[8] H. Yu, Y.-C. Tam, T. Schaaf, S. Stüker, Q. Jin, M. Noamany, and T. Schultz, "The ISL RT04 Mandarin Broadcast News Evaluation System," in *EARS Rich Transcription Workshop*, 2004.

[9] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[10] M. Heck, K. Kubo, M. Sperber, S. Sakti, S. Stker, C. Saam, K. Kilgour, C. Mohr, G. Neubig, T. Toda, S. Nakamura, and A. Waibel, "The KIT-NAIST (contrastive) english ASR system for IWSLT 2012," in *Proceedings of the International Workshop on Speech Translation (IWSLT 2012)*, Hong Kong, December 2012.

[11] T. Kaukoranta, P. Fränti, and O. Nevalainen, "Iterative split-and-merge algorithm for VQ codebook generation," *Optical Engineering*, vol. 37, no. 10, pp. 2726–2732, 1998.

[12] M. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[13] A. Black, P. Taylor, R. Caley, and R. Clark, "The festival speech synthesis system," 1998.

[14] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Commun.*, vol. 50, no. 5, pp. 434–451, May 2008. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2008.01.002

[15] Daniel Povey, Dimitri Kanevsky, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Karthik Visweswariah, "Boosted mmi for model and feature-space discriminative training," in *ICASSP 2008*, 2008, pp. 4057–4060.

[16] Bahl L.R., Brown P.F, de Souza P.V., and L.R. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *ICASSP 1986*, 1986, pp. 49–52.

[17] V. Valtchev, J. J. Odell, P.C. Woodland, and S.J. Young, "MMIE training of large vocabulary recognition systems," in *Speech Communication 22*, 1997, pp. 303–314.

[18] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Seventh International Conference on Spoken Language Processing*, 2002.

[19] T. Marek, "Analysis of german compounds using weighted finite state transducers," *Bachelor thesis, University of Tübingen*, 2006.

[20] M. Schröder and J. Trouvain, "The german text-to-speech synthesis system mary: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003.

[21] H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *Automatic Speech Recognition and Understanding, 2001. ASRU '01. IEEE Workshop on*, 2001, pp. 214–217.

[22] Sebastian Stüker, Christian Fügen, Susanne Burger, and Matthias Wölfel, "Cross-System Adaptation and Combination for Continuous Speech Recognition: The Influence of Phoneme Set and Acoustic Front-End," in *Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech 2006, ICSLP)*, Pittsburgh, PA, USA: ISCA, Nov. 2006, pp. 521–524.

[23] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proceedings the IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, CA, USA, Dec. 1997, pp. 347–354.

Polish - English Speech Statistical Machine Translation Systems for the IWSLT 2013.

Krzysztof Wołk, Krzysztof Marasek

Multimedia Department
Polish Japanese Institute of Information Technology, Koszykowa 86, 02-008 Warsaw
kwolk@pjwstk.edu.pl, kmarasek@pjwstk.edu.pl

Abstract

This research explores the effects of various training settings from Polish to English Statistical Machine Translation system for spoken language. Various elements of the TED parallel text corpora for the IWSLT 2013 evaluation campaign were used as the basis for training of language models, and for development, tuning and testing of the translation system. The BLEU, NIST, METEOR and TER metrics were used to evaluate the effects of data preparations on translation results. Our experiments included systems, which use stems and morphological information on Polish words. We also conducted a deep analysis of provided Polish data as preparatory work for the automatic data correction and cleaning phase.

1. Introduction

Polish is one of the most complex West-Slavic languages, which represents a serious challenge to any SMT system. The grammar of the Polish language, with its complicated rules and elements, together with a big vocabulary (due to complex declension) are the main reasons for its complexity. Furthermore, Polish has 7 cases and 15 gender forms for nouns and adjectives, with additional dimensions for other word classes.

This greatly affects the data and data structure required for statistical models of translation. The lack of available and appropriate resources required for data input to SMT systems presents another problem. SMT systems should work best in specified, not too wide text domains and will not perform well for general use. Good quality parallel data, especially in a required domain has low availability. In general, Polish and English differ also in syntax. English is a positional language, which means that the syntactic order (the order of words in a sentence) plays a very important role, particularly due to limited inflection of words (e.g. lack of declension endings). Sometimes, the position of a word in a sentence is the only indicator of the sentence meaning. In the English sentence, the subject group comes before the predicate, so the sentence is ordered according to the Subject-Verb-Object (SVO) schema. In Polish, however, there is no specific word order imposed and the word order has no decisive influence on the understanding of the sentence. One can express the same thought in several ways, which is not possible in English. For example, the sentence „I bought myself a new car." can be written in Polish as „Kupiłem sobie nowy samochód", or "Nowy samochód sobie kupiłem.", or "Sobie kupiłem nowy samochód.", or „Samochód nowy sobie kupiłem." Differences in potential sentence orders make the translation process more complex, especially when working on a phrase-model with no additional lexical information.

As a result the progress in the development of SMT systems for Polish is substantially slower as compared to other languages. The aim of this work is to create an SMT system for translation from Polish to (and the other way round, i.e. from English to Polish) to address the IWSLT 2013 [2] evaluation campaign requirements. This paper is structured as follows: Section 2 explains the Polish data preparation. Section 3 presents the English language issues. Section 4 describes the translation evaluation methods. Section 5 discusses the results. Sections 6 and 7 summarize potential implications and future work.

2. Preparation of the Polish data

The Polish data in the TED talks (about 15 MB) include almost 2 million words that are not tokenized. The transcripts themselves are provided as pure text encoded with UTF-8 and the transcripts are prepared by the IWSLT team [3]. In addition, they are separated into sentences (one per line) and aligned in language pairs.

It should be emphasized that both automatic and manual preprocessing of this training information was required. The extraction of the transcription data from the provided XML files ensured an equal number of lines for English and Polish. However, some of the discrepancies in the text parallelism could not be avoided. These discrepancies are mainly repetitions of the Polish text not included in the English text.

Another problem is that TED 2013 data is full of errors. Let us first take spelling errors that artificially increase the dictionary size and make the statistics worse. We took a very large Polish dictionary [23] that consists of 2,532,904 different words. Then, we created a dictionary from TED 2013 data and it consisted of 92,135 unique words. Intersection of those 2 dictionaries resulted in a new dictionary containing 58,393 words. It means that in TED 2013 we found 33742 words that do not exist in Polish (spelling errors or named entities). This is as much as 36.6% of the whole TED Polish vocabulary.

To verify that, we conducted a manual analysis on a sample of the first 300 lines from the TED corpora. We found that there were 4268 words containing a total of 35 kinds of spelling errors that occurred many times. But what we found to be more problematic was that there were sentences with odd nesting, such as:

Part A, Part A, Part B, Part B.
e.g.

"Ale będę starał się udowodnić, że mimo złożoności, Ale będę starał się udowodnić, że mimo złożoności, istnieją pewne rzeczy pomagające w zrozumieniu. istnieją pewne rzeczy pomagające w zrozumieniu."

We can see that some parts (words or full phrases or even whole sentences) were duplicated. Furthermore, there are segments containing repeated whole sentences inside one segment. For instance:

Sentence A. Sentence A.
e.g.

“Zakumulują się u tych najbardziej pijanych i skąpych. Zakumulują się u tych najbardziej pijanych i skąpych.”

or:

Part A, Part B, Part B, Part C
e.g.

” Matka może się ponownie rozmnażać, ale jak wysoką cenę płaci, przez akumulację toksyn w swoim organizmie - przez akumulację toksyn w swoim organizmie - śmierć pierwszego młodego.”

We identified 51 out of 300 segments that were mistaken in such way. Overall, in the sample test set we found that we got about 10% of spelling errors and about 17% of insertion errors. However, it must be noted that we simply took the first 300 lines, but in the whole text there are places where more problems occur. So, to some extent, this confirms that there are problems related to the dictionary.

Additionally, there are a number of English names, words and phrases (nor translated) present in the Polish text. There are also some sentences originating from different languages (e.g., German and French). Additionaly some translations are just incorrect or too indirect with not enough precision in translation, e.g. “And I remember there sitting at my desk thinking, Well, I know this. This is a great scientific discovery. ” was translated into “ Pamiętam, jak pomyślałem: To wyjątkowe, naukowe odkrycie.” And the correct translation would be “Pamiętam jak siedząc przy biurku pomyślałem, dobrze, wiem to. To jest wielkie naukowe odkrycie”.

The size of the vocabulary is 92,135 Polish unique words and 41,684 English unique words. The disproportionate vocabulary sizes are also a challenge especially in translation from English to Polish.

Another serious problem (especially for Statistical Machine Translation) that we found was that English sentences were translated in an improper manner.

There were four main problems:

1. Repetitions – when part of the text is repeated several times after translation, i.e.
    a. EN: Sentence A. Sentence B.
    b. PL: Translated Sentence A. Translated Sentence B. Translated Sentence B. Translated Sentence B.
2. Wrong usage of words – when one or more words used for the Polish translation change slightly the meaning of the original English sentence, i.e.
    a. EN: We had these data a few years ago.
    b. PL (the proper meaning of the polish sentence): We’ve been delivered these data a few years ago.
3. Indirect translations, usage of metaphors – when the Polish translation uses different wording in order to preserve the meaning of the original sentence, especially when the exact translation would result in a sentence with no sense. Many metaphors are translated this way.
4. Translations that are not precise enough – when the translated fragment does not contain all the details of the original sentence, but only its overall meaning is the same.

Looking at the style of the translated text, it can be concluded that the text was translated by several people who translated it independently of one another. The text was divided between them and then the fragments of it were merged into one, thus:

- In some places, the text looks as if it was not translated by a human, but by an automatic system instead.
- Some paragraphs contain a lot of metaphors, which will certainly interfere with the subsequent translation. The translations are not direct.
- We also found some problems with encoding of Polish characters and with usage some strange symbols in the text ex. ☋, Ψ, ∑, ☒, ת, □, etc.

Another problem is that the TED Talks do not have any specific domain. Statistical Machine Translation by definition works best when very specific domain data is used. The data we have is a mix of various, unrelated topics. This is most likely the reason why we cannot expect big improvements with this data.

There is not much focus on Polish in the campaign, so there is almost no data in Polish in comparison to a huge amount of data in, for example, French or German. What is more, provided Polish samples are not only small, but also in a different domain, which does not enrich a required language model well enough. At first we used perplexity measurement metrics to determine the data we got. Some of it we were able to obtain from the project page, some from another project and the rest was collected manually using web crawlers. We created those corpora and used them according to the permission from organizers [22]. What we created was:

- A Polish – English dictionary (bilingual parallel)
- Additional (newer) TED Talks data sets not included in the original train data (we crawled bilingual data and created a corpora from it) (bilingual parallel)
- E-books (monolingual PL + monolingual EN)
- Euro News Data (bilingual parallel)
- Proceedings of UK Lords (monolingual EN)
- Subtitles for movies and TV series (monolingual PL)
- Parliament and senate proceedings (monolingual PL)

“Other” in the table below stands for many very small models merged together. We show here the perplexity values and the perplexity values with no smoothing (PPL in Table I) of those language models smoothed with the Kneser-Ney algorithm (PPL+KN in Table I). We used the MITLM toolkit for that

evaluation. As an evaluation set we used dev2010 data, which was used for tuning. Its dictionary covers 2861 different words.

Table 1: Data Perplexities for dev2010 data set

Data set	Dictionary	PPL	PPL + KN
Baseline train.en	44,052	221	223
EMEA	30,204	1738	1848
KDE4	34,442	890	919
ECB	17,121	837	889
OpenSubtitles	343,468	388	415
EBOOKS	528,712	405	417
EUNEWS	34,813	430	435
NEWS COMM	62,937	418	465
EUBOOKSHOP	167,811	921	950
UN TEXTS	175,007	681	714
UK LORDS	215,106	621	644
NEWS 2010	279,039	356	377
GIGAWORD	287,096	582	610
DICTIONARY	39,214	8629	8824
OTHER	13,576	492	499
TEDDL	47,015	277	277

EMEA are texts from the European Medicines Agency, KDE4 is a localization file of that user GUI, ECB stands for European Central Bank corpus, OpenSubtitles are movies and TV series subtitles, EUNEWS is a web crawl of the euronews.com web page and EUBOOKSHOP comes from bookshop.europa.eu. Lastly bilingual TEDDL is additional TED data. As can be seen from the table above, all additional data is much worse than the files provided in the baseline system, so no major improvements based only on data could be anticipated.

Before the use of a training translation model, preprocessing that included removal of long sentences (set to 80 words) had to be performed. The Moses toolkit scripts[6] were used for this purpose. Moses is an open-source toolkit for statistical machine translation which supports linguistically motivated factors, confusion network decoding, and efficient data formats for translation models and language models. In addition to the SMT decoder, the toolkit also includes a wide variety of tools for training, tuning and applying the system to many translation tasks. In addition, the text in the TED data set had to be repaired in a number of ways to correct spelling errors and grammar errors, ensure that there was only one sentence on each line, remove language translations that were not of interest, remove HTML and XML tags within text, remove of strange symbols not existing in a specific language and repetitions of words and sentences.

The final processing included 134,678 lines from the Polish to English corpus. However, the disproportionate vocabulary sizes remained, with 41,163 English words and 92,135 Polish words. One of the solutions to this problem (according to work of Bojar [7]) was to use stems instead of surface forms that reduced the Polish vocabulary size to 40,346. Such a solution also requires a creation of an SMT system from Polish stems to plain Polish. Subsequently, morphosyntactic tagging, using the Wroclaw Natural Language Processing (NLP) tools (nlp.pwr.wroc.pl), was included as an additional information source for the SMT system preparation. It can be also used as a first step for implementing a factored SMT system that, unlike a phrase-based system, includes morphological analysis, translation of lemmas and features as well as generation of surface forms. Incorporating additional linguistic information should effectively improve translation performance [8].

2.1. Polish stem extraction

As previously mentioned, stems extracted from Polish words are used instead of surface forms to overcome the problem of the huge difference in vocabulary sizes. Keeping in mind that in half of the experiments the target language was English in the form of normal sentences, it was not necessary to introduce models for converting the stems to the appropriate grammatical forms; however it will be part of our future work in translation into Polish. For Polish stem extraction, a set of natural language processing tools available at http://nlp.pwr.wroc.pl was used [9]. These tools can be used for:

- Tokenization
- Morphosyntactic analysis
- Shallow parsing as chunking
- Text transformation into the featured vectors

The following two components are also used:

- MACA –a universal framework used to connect the different morphological data
- WCRFT – this framework combines conditional random fields and tiered tagging

These tools used in sequence provide an XML output. It includes a surface form of the tokens, stems and morphosyntatic tags. An example of such data is given in section 2.2.

2.2. Morphosyntactic element tagging with standard tools

Wroclaw's tools were used to tag morphosyntactic elements. More precise tagging can be achieved with these settings. In addition, every tag in this tagset consists of specific grammatical classes with specific values for particular attributes. Furthermore, these grammatical classes include attributes with values that require additional specification. For example, nouns require numbers while adverbs require an appropriate degree of an attribute. This causes segmentation of the input data, including tokenization of words in a different way as compared to the Moses tools. On the other hand, this causes problems with building parallel corpora. This can be solved by placing markers at the end of input lines.

In the following example, where pl.gen. "men" is derived from sin.nom."człowiek" (*man*) or pl.nom. "ludzie" (*people*), it can be demonstrated how one tag is used where, in the most difficult cases, more possible tags are provided.

```
<tok>
<orth>ludzi</orth>
<lex disamb="1"><base>człowiek</base>
<ctag>subst:pl:gen:m1</ctag></lex>
<lex disamb="1"> <base>ludzie</base>
<ctag>subst:pl:gen:m1</ctag></lex>
</tok>
```

In this example, only one form (the first stem) is used for further processing.

We developed an XML extractor tool to generate three different corpora for the Polish language data:

- Words in the infinitive form
- Subject-Verb-Object (SVO) word order
- both the infinitive form and the SVO word order

This allows experiments with those preprocessing techniques.

Moreover, some of the NLP tools use the Windows-1250 Eastern Europe Character Encoding, which requires a conversion of information to and from the UTF-8 encoding that is commonly used in other, standard tools.

3. English Data Preparation

The preparation of the English data was definitively less complicated than for Polish. We developed a tool to clean the English data by removing foreign words, strange symbols, etc. Compare to polish english data contained significantly less errors. Nonetheless some problems needed to be removed, most problematic were translations into languages other than english, strange UTF-8 symbols. We also found few duplications and insertions inside single segment.

4. Evaluation Methods

Metrics are necessary to measure the quality of translations produced by the SMT systems. For this, various automated metrics are available to compare SMT translations to high quality human translations. Since each human translator produces a translation with different word choices and orders, the best metrics measure SMT output against multiple reference human translations. Among the commonly used SMT metrics are: Bilingual Evaluation Understudy (BLEU), the U.S. National Institute of Standards & Technology (NIST) metric, the Metric for Evaluation of Translation with Explicit Ordering (METEOR), and Translation Error Rate (TER). These metrics will now be briefly discussed. [10]

BLEU was one of the first metrics to demonstrate high correlation with reference human translations. The general approach for BLEU, as described in [9], is to attempt to match variable length phrases to reference translations. Weighted averages of the matches are then used to calculate the metric. The use of different weighting schemes leads to a family of BLEU metrics, such as the standard BLEU, Multi-BLEU, and BLEU-C. [11]

As discussed in [11], the basic BLEU metric is:

$$BLEU = P_B \exp\left(\sum_{n=0}^{N} w_n \log p_n\right)$$

where p_n is an *n*-gram precision using *n*-grams up to length N and positive weights w_n that sum to one. The brevity penalty P_B is calculated as:

$$P_B = \begin{cases} 1, & c > r \\ e^{(1-r/c)}, & c \leq r \end{cases}$$

where *c* is the length of a candidate translation, and *r* is the effective reference corpus length. [9]

The standard BLEU metric calculates the matches between *n*-grams of the SMT and human translations, without considering position of the words or phrases within the texts. In addition, the total count of each candidate SMT word is limited by the corresponding word count in each human reference translation. This avoids bias that would enable SMT systems to overuse high confidence words in order to boost their score. BLEU applies this approach to texts sentence by sentence, and then computes a score for the overall SMT output text. In doing this, the geometric mean of the individual scores is used, along with a penalty for excessive brevity in translation. [9]

The NIST metric seeks to improve the BLEU metric by valuing information content in several ways. It takes the arithmetic versus geometric mean of the *n*-gram matches to reward good translation of rare words. The NIST metric also gives heavier weights to rarer words. Lastly, it reduces the brevity penalty when there is a smaller variation in translation length. This metric has demonstrated that these changes improve the baseline BLEU metric. [12]

The METEOR metric, developed by the Language Technologies Institute of Carnegie Mellon University, is also intended to improve the BLEU metric. METEOR rewards recall by modifying the BLEU brevity penalty, takes into account higher order *n*-grams to reward matches in word order, and uses arithmetic vice geometric averaging. For multiple reference translations, METEOR reports the best score for word-to-word matches. Banerjee and Lavie [13] describe this metric in detail.

As found in [13], this metric is calculated as follows:

$$METEOR = \left(\frac{10\,P\,R}{R + 9\,P}\right)(1 - P_M)$$

where *P* is the unigram precision and *R* is the unigram recall. The METEOR brevity penalty P_M is:

$$P_M = 0.5\left(\frac{C}{M_U}\right)$$

where *C* is the minimum number of chunks such that all unigrams in the machine translation are mapped to unigrams in the reference translation. M_U is the number of unigrams that matched.

The METEOR metric incorporates a sophisticated word alignment technique that works incrementally. Each alignment stage attempts to map previously unmapped words in the SMT and reference translations. In the first phase of each stage, METEOR attempts three different types of word-to-word mappings, in the following order: exact matches, matches using stemming, and matches of synonyms. The second stage uses the resulting word mappings to evaluate word order similarity. [13]

Once a final alignment of the texts is achieved, METEOR calculates precision similar to the way the NIST metric calculates it. METEOR also calculates word-level recall between the SMT translation and the references, and combines this with precision by computing a harmonic mean that weights recall higher than precision. Lastly, METEOR penalizes shorter *n*-gram matches and rewards longer matches. [13]

TER is one of the most recent and intuitive SMT metrics developed. This metric determines the minimum number of human edits required for an SMT translation to match a reference translation in meaning and fluency. Required human edits might include inserting words, deleting words, substituting words, and changing the order or words or phrases. [14]

5. Experimental Results

A number of experiments were performed to evaluate various versions for our SMT systems. The experiments involved a number of steps. Processing of the corpora was accomplished, including tokenization, cleaning, factorization, conversion to lower case, splitting, and a final cleaning after splitting. Training data was processed, and the language model was developed. Tuning was performed for each experiment. Lastly, the experiments were conducted.

The baseline system testing was done using the Moses open source SMT toolkit with its Experiment Management System (EMS) [15]. The SRI Language Modeling Toolkit (SRILM) [16] with an interpolated version of the Kneser-Key discounting (interpolate –unk –kndiscount) was used for 5-gram language model training. We used the MGIZA++ tool for word and phrase alignment. KenLM [19] was used to binarize the language model, with a lexical reordering set to use the msd-bidirectional-fe model. Reordering probabilities of phrases are conditioned on lexical values of a phrase. It considers three different orientation types on source ant target phrases like monotone(M), swap(S) and discontinuous(D). The bidirectional reordering model adds probabilities of possible mutual positions of source counterparts to current and following phrases. Probability distribution to a foreign phrase is determined by "f" and to the English phrase by "e" [20,21]. MGIZA++ is a multi-threaded version of the well-known GIZA++ tool [17]. The symmetrization method was set to grow-diag-final-and for word alignment processing. First two-way direction alignments obtained from GIZA++ were intersected, so only the alignment points that occured in both alignments remained. In the second phase, additional alignment points existing in their union were added. The growing step adds potential alignment points of unaligned words and neighbors. Neighborhood can be set directly to left, right, top or bottom, as well as to diagonal (grow-diag). In the final step, alignment points between words from which at least one is unaligned are added (grow-diag-final). If the grow-diag-final-and method is used, an alignment point between two unaligned words appears. [18]

We conducted about three hundred of experiments to determine the best possible translation from Polish to English and the reverse. For experiments we used Moses SMT with Experiment Management System (EMS) [24]. Starting from baseline (BLEU: 16,02) system tests, we raised our score through extending the language model with more data and by interpolating it linearly. Firstly we used OpenSubtitles bilingual corpora for training and raised the BLEU score to 17,71. In the next step, we interpolated OpenSubtitles language model with original one instead of merging them. We determined that the linear interpolation gives better results than the log-linear one, when using our data. In the next steps, we interpolated some data and also added a Polish-English dictionary. This produces BLEU score equal to 20,41. In the PL->EN experiment number 170th we managed to determine better settings for the language model. We set the order from 5 to 6 and changed the discounting method from Kneser-Ney to Witten-Bell. In the training part, we changed the reordering method from msd-bidirectional-fe to msd-fe. For now, it produces the best score we were able to obtain (20,88).

As previously described, we also tried to work with stems, but the results weren't satisfying enough. Scores in fact were a bit lower – most likely because there were errors in texts. The Wroclaw NLP tools, when given a text with errors, produces even more errors and we lose some parts of the data and good alignment, which we assume is the reason for the worse score. Nevertheless, it is worth to give it a look in future research.

Because of a much bigger dictionary, the translation from EN to PL is significantly more complicated. We also lacked the data. Our baseline system score was 8,49 in BLEU. First, we tried working with stems by changing data to infinitives and reordering parts of sentences into SVO form. We then interpolated a language model containing e-books (it was prepared by us) and raised the score a little higher (9,42). Preparation of other language models and adding a bit more data resulted in achieving better scores. We also increased the n-gram order to 6, which produced BLEU score of 10,27. Next, we started to add train data (dictionary raised score to 10,40 and OpenSubtitles to 10,49) changing the alignment symmetrization method from msd-bidirectional-fe to tgttosrc (target to source) and obtained a slightly higher score. Lastly, we raised the max sentence length from 80 to 90 and achieved the highest score so far, which was 10,68 in BLEU. It must be noted that in order for the Wroclaw NLP tools to work correctly all data sets had to be previously cleaned by our tool in order to retain good alignment.

The experiments, conducted with the use of the test data from years 2010-2013, are defined in Table 1 and Table 2, respectively, for the Polish-to-English and English-to-Polish translations. They are measured by the BLEU, NIST, TER and METEOR metrics. Note that a lower value of the TER metric is better, while the other metrics are better when their values are higher. BASE stands for baseline system with no improvements, COR is a system with corrected spelling in Polish data, INF is a system using infinitive forms in Polish, SVO is a system with the subject – verb – object word order in a sentence and BEST stands for the best result we achieved.

Table 2: Polish-to-English translation

System	Year	BLEU	NIST	TER	METEOR
BASE	2010	16.02	5.28	66.49	49.19
COR	2010	16.09	5.22	67.32	49.09
BEST	2010	20.88	5.70	64.39	52.74
INF	2010	13.22	4.74	70.26	46.30
SVO	2010	9,29	4,37	76,59	43,33
BASE	2011	18.86	5.75	62.70	52.72
COR	2011	19.18	5.72	63.14	52.88
BEST	2011	23.70	6.20	59.36	56.52
BASE	2012	15.83	5.26	66.48	48.60
COR	2012	15.86	5.32	66.22	49.00
BEST	2012	20.24	5.76	63.79	52.37
BASE	2013	16.55	5.37	65.54	49.99
COR	2013	16.98	5.44	65.40	50.39
BEST	2013	23.00	6.07	61.12	55.16
INF	2013	12.40	4.75	70.38	46.36

Table 3: English-to-Polish translation

System	Year	BLEU	NIST	TER	METEOR
BASE	2010	8.49	3.70	76.39	31.73
COR	2010	9.39	3.96	74.31	33.06
BEST	2010	10.72	4.18	72.93	34.69
INF	2010	9.11	4.46	74.28	37.31
SVO	2010	4,27	4,27	76,75	33,53
BASE	2011	10.77	4.14	71.72	35.17
COR	2011	10.74	4.14	71.70	35.19
BEST	2011	15.62	4.81	67.16	39.85
BASE	2012	8.71	3.70	78.46	32.50
COR	2012	8.72	3.70	78.57	32.48
BEST	2012	13.52	4.35	73.36	36.98
BASE	2013	9.35	3.69	78.13	32.52
COR	2013	9.35	3.70	78.10	32.54
BEST	2013	14.37	4.42	72.06	37.87
INF	2013	13.30	4.83	70.50	35.83

Official results were obtained late for this paper publication so we decided only to put our best translation system results. In translation from Polish to English, case-sensitive BLEU score was 22,60 and TER 62,56 while in case-insensitive BLEU score was equal to 23,54 and TER 61,12. In translation from English to Polish we obtained case-sensitive BLEU 14,29 and TER 73,53 while case-insensitive scores were 15,04 for BLEU and 72,05 for TER.

6. Discussion

Several conclusions can be drawn from the experimental results presented here. Automatic and manual cleaning of the training files has some impact, among the variations examined, on improving translation performance, together with spelling correction of the data in Polish – although it resulted in better BLEU and METEOR scores, not always in higher NIST or TER metrics. In particular, automatic cleaning and conversion of verbs to their infinitive forms improved translation performance when it comes to the English-to-Polish translation, quite the contrary to the Polish-to-English translation. This is likely due to a reduction of the Polish vocabulary size. Changing the word order to SVO is quite interesting. It didn't help at all in some cases, although one would expect it to. When it comes to experiments from PL to EN, the score was always worse, which was not anticipated. On the other hand, in the EN to PL experiments in some cases improvement could be seen. Although the BLEU score dramatically decreased and TER became slightly worse, NIST and METEOR showed better results than the baseline system. Most likely it is because of each metric has different evaluation method. The BLEU and TER scored decreased probably because phrases were mixed in the SVO conversion process. It is worth investigating (especially the PL to EN system). Maybe there was some kind of implementation error in our parser or cleaner.

In summary converting Polish verbs to infinitives reduces the Polish vocabulary, which should improve the English-to-Polish translation performance. The Polish to English translation typically outscores the English to Polish translation, even on the same data. This requires further evaluation.

7. Conclusion and future work

Several potential opportunities for future work are of interest. Additional experiments using extended language models are warranted to determine if this improves SMT scores. We are also interested in developing some more web crawlers in order to obtain additional data that would most likely prove useful.

Currently, neural network based language models are amongst the most successful techniques for statistical language modeling. They can be easily applied in a wide range of tasks, including automatic speech recognition and machine translation and they also provide significant improvements over classic backoff n-gram models. The 'rnnlm' toolkit can be used in order to train, evaluate and use such models. With the RNNLM toolkit we were able to reduce perplexity (Table 4) a little. We intend to explore it more in future work.

Table 4: RNNLM Results

TOOLKIT	PPL
RNNLM	193.31
SRILM	216.79
Combination of RNNLM and SRILM	169.55

The test was conducted on a default language model and a test set provided in TED 2013 data, and it looks promising. The language model vocabulary was 44052 words and the test file was 3575 words.

Polish is a language that has a complex grammar, which is why it is very hard to translate from and into languages of lower complexity such as English. Creating a factored model for SMT would most probably improve its performance. We are planning on implementing an SMT factored system based on POS tags.

An ideal SMT system should be fully automatic. To use infinitives, we will have to make this conversion automatic with usage of Wroclaw NLP tools. Lastly, it is our objective to create two SMT systems, one converting Polish words to Polish stems (and vice versa), and another converting Polish infinitives to English in order to make translations fully automatic.

The observed lower quality of the translation system based on conversion into an SVO sentence form requires further investigation.

8. Acknowledgements

This work is supported by the European Community from the European Social Fund within the Interkadra project UDA-POKL-04.01.01-00-014/10-00 and Eu-Bridge 7th FR EU project (grant agreement n°287658).

References

[1] Russell, S.J. and Norvig, P. Artificial Intelligence: A Modern Approach, Third Edition, Upper Saddle R: Prentice Hall, 2010, pp. 907-910.

[2] IWSLT2013 Evaluation Campaign, www.iwslt2013.org

[3] http://www.iwslt2013.org/59.php

[4] The EuroMatrix Project, http://www.euromatrix.net/

[5] OPUS Subtitles Corpus, http://opus.lingfil.uu.se/

[6] Koehn P., Hoang H., Birch A., Callison-Burch C., Federico M., Bertoldi N., Cowan B., Shen W., Moran C., Zens R., Dyer C., Bojar R., Constantin A., Herbst E., Moses: Open Source Toolkit for Statistical Machine Translation, Proceedings of the ACL 2007 Demo and Poster Sessions, pages 177–180, Prague, June 2007

[7] Marasek, K., "TED Polish-to-English translation system for the IWSLT 2012", Proc. of International Workshop on Spoken Language Translation (IWSLT) 2010, Hong Kong, December 2012.

[8] Bojar O., "Rich Morphology and What Can We Expect from Hybrid Approaches to MT". Invited talk at International Workshop on Using Linguistic Information for Hybrid Machine Translation(LIHMT-2011), http://ufal.mff.cuni.cz/~bojar/publications/2011-FILEbojar_lihmt_2011_pres-PRESENTED.pdf, 2011

[9] Amittai E. Axelrod, Factored Language Models for Statistical Machine Translation, University of Edinburgh, 2006

[10] Radziszewski A., "A tiered CRF tagger for Polish", in: Intelligent Tools for Building a Scientific Information Platform: Advanced Architectures and Solutions, editors: Membenik R., Skonieczny L., Rybiński H., Kryszkiewicz M., Niezgódka M., Springer Verlag, 2013 (to appear)

Radziszewski A., Śniatowski T., "Maca: a configurable tool to integrate Polish morphological data", Proceedings of the Second International Workshop on Free/OpenSource Rule-Based Machine Translation, FreeRBMT11, Barcelona, 2011

[11] Philipp Koehn, What is a Better Translation? Reflections on Six Years of Running Evaluation Campaigns, 2011

[12] Papineni, K., Rouskos, S., Ward, T., and Zhu, W.J. "BLEU: a Method for Automatic Evaluation of Machine Translation", Proc. of 40th Annual Meeting of the Assoc. for Computational Linguistics, Philadelphia, July 2002, pp. 311-318.

[13] Doddington, G., "Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics", Proc. of Second International Conference on Human Language Technology (HLT) Research 2002, San Diego, March 2002, pp. 138-145.

[14] Banerjee, S. and Lavie, A., "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments", Proc. of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, June 2005, pp. 65-72.

[15] Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J., "A Study of Translation Edit Rate with Targeted Human Annotation", Proc. of 7th Conference of the Assoc. for Machine Translation in the Americas, Cambridge, August 2006.

[16] Koehn, P. et al., "Moses: Open Source Toolkit for Statistical Machine Translation," Annual Meeting of the Association for Computational Linguistics (ACL) demonstration session, Prague, June 2007.

[17] Stolcke, A., "SRILM – An Extensible Language Modeling Toolkit", INTERSPEECH, 2002.

[18] Gao, Q. and Vogel, S., "Parallel Implementations of Word Alignment Tool", Software Engineering, Testing, and Quality Assurance for Natural Language Processing, pp. 49-57, June 2008.

[19] http://www.cs.jhu.edu/~ccb/publications/iwslt05-report.pdf

[20] Heafield, K. "KenLM: Faster and smaller language model queries", Proc. of Sixth Workshop on Statistical Machine Translation, Association for Computational Linguistics, 2011.

[21] Marta R. Costa-jussa, Jose R. Fonollosa, Using linear interpolation and weighted reordering hypotheses in the Moses system, Barcelona, Spain, 2010

[22] Shared corporas: www.korpusy.s16874487.onlinehome-server.info and approvements https://groups.google.com/forum/#!topic/iwslt-e/TiQH5jERE5Y

[23] Brocki Ł., Marasek K., Koržinek D., "Multiple Model Text Normalization for the Polish Language", Foundations of Intelligent Systems, 20th International Symposium, ISMIS 2012, Macau, China, December 4-7 2012, pp. 143-148.

[24] http://www.statmt.org/moses/?n=FactoredTraining.EMS

The RWTH Aachen German and English LVCSR systems for IWSLT-2013

M. Ali Basha Shaik[1], *Zoltan Tüske*[1], *Simon Wiesler*[1],
Markus Nußbaum-Thom, Stephan Peitz, Ralf Schlüter[1] *and Hermann Ney*[1,2]

[1]Human Language Technology and Pattern Recognition – Computer Science Department
RWTH Aachen University, 52056 Aachen, Germany
[2]Spoken Language Processing Group, LIMSI CNRS, Paris, France
{shaik, tuske, wiesler, nussbaum, peitz, schlueter, ney}@cs.rwth-aachen.de

Abstract

In this paper, German and English large vocabulary continuous speech recognition (LVCSR) systems developed by the RWTH Aachen University for the IWSLT-2013 evaluation campaign are presented. Good improvements are obtained with state-of-the-art monolingual and multilingual bottleneck features. In addition, an open vocabulary approach using morphemic sub-lexical units is investigated along with the language model adaptation for the German LVCSR. For both the languages, competitive WERs are achieved using system combination.

1. Introduction

This paper describes in detail the German and English RWTH large vocabulary continuous speech recognition recognition systems developed for the IWSLT-2013 evaluation campaign. Automatic speech recognition track in IWSLT-2013 evaluation campaign focuses on transcribing lecture data. One of the major challenge in the IWSLT-2013 evaluation is that no acoustic modeling training data is provided for the aforementioned languages, but the development data. The data includes speech types like lectures, talks and conversations. Recognition on the data is challenging because of a huge variability in the acoustic conditions and a large portion includes spontaneous speech.

In the development of ASR systems transcribed speech data is still a significant cost factor. Therefore, methods which are able to reuse out-of-domain or multilingual resources to ease the model training, have growing interest, and this demand exists not only for under-resourced languages. The neural networks (NN) have become a major component in the state-of-the-art ASR system, and are used to extract features (probabilistic [1] or bottleneck (BN) TANDEM approach [2]) and/or to model the emission probability in the HMM framework directly (hybrid approach) [3]. In [4, 5] it was observed that Multi Layer Perceptron (MLP) based NN posterior features possess language independent properties to a certain degree: the cross-lingual porting of NNs could lead to significant improvement in a different language. In order to exploit resources of multiple languages in acoustic model training, there is usually a need to unify similar sounds across different languages e.g. by IPA or SAMPA. However, as was shown by [6] the training of NNs on multiple languages is possible without such a mapping if language dependent output layers are used and only the hidden layer parameters are shared between the languages. Combining the multilingual learning with the bottleneck approach [7, 8] demonstrated that the multilingual BN features could benefit from the additional non-target language data and outperformed the unilingual BN. Through better generalization the multilingual BN features can offer improved portability on an new language, and acoustical mismatch between the training and testing can be reduced in the target language by exploiting matched data from other languages [9]. Since transcribed lecture data were not provided for the evaluation, in our systems the BN features are trained on large amount of broadcast news and conversations data of multiple languages. Covering wide variety of acoustic conditions through the multilingual resources, we aimed at improving the robustness of the acoustic model to recognize acoustically less matched lecture data. On the other hand, German is a morphologically rich language having a high degree of word inflections, derivations and compounding. For a morphologically rich language like German, high out-of-vocabulary (OOV) rates and poor LM probabilities are generally observed. Thus, sub-lexical language modeling is used to decrease the OOV rate and reduce the data sparsity [10, 11, 12]. In this work, we also investigate the use of the state-of-the-art LMs like Maximum Entropy (MaxEnt) LMs, which provide modular structure to incorporate various knowledge sources as features in the sub-lexical LMs. Furthermore, we experiment the use of Maximum a-posteriori (MAP) adaptation over the MaxEnt LMs. Thus, the benefits of both the MaxEnt LMs and the traditional N-gram backoff LMs are effectively combined using interpolation, followed by confusion network based system combination.

The rest of the paper is organized as follows: In Section 2 speaker independent and dependent acoustic models are described along with the investigated features. In Section 3, the use of various full-word and sub-lexical language models are investigated. In Section 3.7, the generation of the lexicon is described. In Section 4, various recognition setups are described. Results are discussed in Section 5, followed by conclusions.

2. Acoustic Model (AM)

In this work, the data from the Quaero project is used for acoustic modeling. The training data for the IWSLT-2013 evaluation campaign consist of data from three domains. While the majority of the data is from the web (WEB), data from broadcast news (BN) and European parliament plenary sessions (EPPS) is also covered.

2.1. Resources

2.1.1. German

Table 1 lists the amount of audio data used from different domains [13] for German LVCSR . Overall, 140 hours of across-domain acoustic training data is used. The data includes the audio from BN, EPPS and the web domains.

Table 1: *Acoustic Training data (dur.: duration (hours), seg.:segments)*

Corpus	#Dur.	#Segs	# Running words
EPPS08	5	1109	45,796
WEB08	14	3452	127,086
Quaero 2010+2011+2012	123	25061	1,391,468

2.1.2. English

Similarly, Table 2 lists the amount of audio data, which is collected from different domains. Overall, 142 hours of acoustic training data is used [13]. The *HUB4* and the *TDT4* corpora contain only American English Broadcast News, whereas the *TC-STAR* corpus consists of European Planery Parliamentary Speech data.

Table 2: *Acoustic Training data (dur.: duration (hours), seg.:segments)*

Corpus	# Dur.	#Segs	# Running words
Quaero	268	57,629	1,666,733
HUB4	206	119.658	1,617,099
TDT4	186	110.266	1,715,445
EPPS	102	66,670	761,234
TED	200	21,614	1,857,660

Table 2 lists the amount of audio data used for acoustic model training. The largest database is the English Quaero corpus[1], which consists of 268 hours transcribes web podcasts. HUB4 and TDT4 are American English broadcast news corpora. EPPS consists of 102 hours of English European Parliament speeches.

All this data has in common that it is out-of-domain for a lectures recognition system. Therefore, we downloaded 200 hours videos from the TED website[2]. All videos have been uploaded to the TED website before the IWSLT cut-off date December31 2010. We used the video subtitles as transcriptions. We used a low pruning threshold for aligning the data and discarded the segments which could not be aligned. In total, we used 962 hours audio training data with a mix of British and American English and from various domains.

[1] http://www.quaero.org/

[2] www.ted.com

2.2. Feature Extraction

2.2.1. Cepstral features

From the audio files 16 Mel-cepstral coefficients (MFCC) were extracted every 10 ms. The 20 logarithmic critical band energies (CRBE) were computed over a Hanning window of 25 ms. For the piecewise linear vocal tract length normalization (VTLN) text-independent Gaussian mixture classifier was trained to estimate the warping factor (fast-VTLN). After the segment-wise mean and variance normalization, 9 consecutive frames of MFCC were mapped by linear discriminant analysis (LDA) to a 45-dimensional subspace.

2.2.2. Multilingual bottleneck MRASTA features

For both evaluation systems the same multilingual MRASTA features are applied. The original RASTA filters were introduced to extract features which are less sensitive to linear distortion [14]. According to [15], the temporal trajectories of the CRBEs were smoothed by two-dimensional bandpass filters to cover the relevant modulation frequency range (MRASTA). One second trajectory of each critical band is filtered by first and second derivatives of the Gaussian function, where the standard deviation varies between 8 and 60 ms resulting in 12 temporal filters per band. Our final `BN` features are extracted from hierarchical, MLP based processing of the modulation spectrum [16, 17]. The input of the first MLP contains the fast modulation part of the MRASTA filtering, whereas the second MLP is trained on the slow modulation components and the PCA transformed BN output of the first MLP. The modulation features fed to the MLPs were always augmented by the CRBE.

Furthermore, in order to extract robust MLP features a multilingual training method proposed by [6] is applied. The MLP training data covered four languages — English, French, German, and Polish —, and the final multilingual `BN` features are trained on $\sim$ 800 hours of speech data collected within the Quaero project as shown in Table 3. The multilingual corpus incorporates the complete German and part of the English resources described in Subsection 2.1.1 and 2.1.2. The feature vectors extracted from the joint corpus of the four languages were randomized and fed to the MLPs. Using language specific softmax outputs, back propagation is initiated only from the language specific subset of the output depending on the language-ID of the feature vector. The MLPs are trained according to cross-entropy criterion, and approximate 1500 tied-triphone state posterior probabilities per each language [18]. To prevent over-fitting and for adjusting the learning rate parameter, 10% of the training corpus is used for cross-validation.

The BN features of the evaluation systems were based on deep MLP. The size of the 6 non-BN hidden layers was set to 2000, the bottleneck layers consisted of 60 nodes and was always placed before the last hidden layer.

Table 3: *Multilingual broadcast news and conversation resources used for* BN *feature training.*

language	German	English	French	Polish
Amount of speech [h]	142	232	317	110

In addition, four additional experiments were carried out to select the best MLP features for German LVCSR : In the classical (shallow) 5-layer uni- and multilingual BN networks the hidden layers had 7000 nodes. In deep BN, making the last hidden layer language dependent (4x2000) increased the number of trainable parameters and did not increase the MLP training time. On the contrary, testing a single large hidden layer (8000 nodes) after the BN increased the number of parameters even further, and resulted in longer training time. The final submissions are based on this later BN structure, one level of the hierarchy is also shown in Fig. 1.

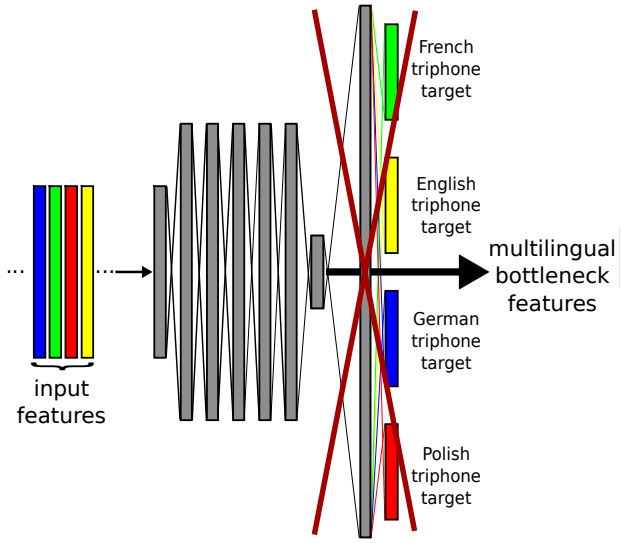


Figure 1: *The joint training of deep context-dependent bottleneck MLP features on multiple languages (FR, EN, DE, PL). The different colors indicate different languages, and language dependent back-propagation from the output layer. The other parts of the network including the bottleneck layer are shared between the languages.*

2.3. AM Training with Speaker Adaptation

The English acoustic models have been trained on the complete data as described in Subsection 2.1.2, whereas the German acoustic models are built using mostly Quaero data as described in Subsection 2.1.1.

All our systems are based on a bottleneck tandem approach, i.e., the outputs of a neural network are used as input features for a Gaussian mixture model (GMM). The final 83-dimensional feature vectors were obtained by concatenating the spectral features with the multi-layer-perceptron (MLP) features described in 2.2.1. The acoustic models AM training followed similar recipes, the GMMs have been trained according to the maximum likelihood (ML) criterion with the expectation maximization algorithm (EM) with Viterbi approximation and a splitting procedure. The GMMs have a globally pooled, diagonal covariance matrix. $4,500$ generalized triphones determined by a decision-tree-based clustering (CART) are modeled in both languages.

Speaker adaptation is of crucial importance for the performance of a lecture recognition system. If significant amount of audio data is available along with the speaker related information, this helps to capture the speaker variabilities and helps in reduction of the WER. Several speaker adaptation techniques are used in our system. First, mean and variance normalization has been applied to the spectral features. Furthermore, we applied a vocal tract length normalization (VTLN) to the MFCC features. The VTLN warping factors were obtained by performing a grid search on the audio training data. A Gaussian classifier has been trained on the results and applied to the training and recognition data to obtain the VTLN-transformed features. In addition, speaker adaptation using constrained maximum likelihood linear regression (CMLLR) [19] with the simple target model approach [20] is applied. The CMLLR transformation is applied to the training data and a new GMM is trained (speaker adaptive training). In recognition, the CMLLR transforms are estimated from a first recognition pass and then, a second recognition pass with the GMM from speaker adaptive training (SAT) is performed. The speaker labels required for for CMLLR adaptation were obtained by clustering speech segments optimizing the Bayesian information criterion [21]. Both the speaker independent and adaptive GMM models ended up over 1M densities. This is referred as common system for both English (system-1) and German LVCSRs.

In addition to the system described above, for English a second system (system-2) is trained which uses the MLP features of our IWSLT-12 submission [22]. These MLPs were only trained on the English Quaero data and have less layers. In order to improve system variability, we also performed an additional recognition pass with maximum likelihood linear regression (MLLR) [19]. In our experience, MLLR does not improve performance of Tandem systems, but it may be advantageous to have an MLLR system in the system combination.

3. Language Model

3.1. Resources

The distribution of words in any spoken language is captured by the LM text. The LM text is collected from various domains. Relatively as more amount of acoustic training data is available for BN than for EPPS and since the BN domain could be closer to the web domain than parliamentary speeches, we decide to build an American English BN AM and a British English EPPS AM in order to get better domain dependent modeling. For the training of the LM we apply a

similar approach, as domain dependent LM data is used. The text is normalized using language dependent predefined set of rules and semi-automatic methods. For example, Dates and Roman numerals are converted into text format. Punctuation's are discarded. In this paper, LM text is used for both the German and English LVCSR task as recommended by the IWSLT evaluation committee[3], as shown in Table 4

Table 4: *Text Resources for German and English LVCSR*

Lang	Corpus	# Running words
DE	Podcast	46k
	IWSLT LM data	2.5M
	Lecture Talks	2.5M
	CALL HOME - speech	5.9M
	Multilingual Parallel data	104M
	Web	384M
	News + acoustic trans.	971M
EN	IWSLT LM data	3M
	WMT 2012 news-commentary	5M
	Acoustic transcriptions	8M
	WMT 2012 news-crawl	2.8B
	Gigaword corpus	3B

3.2. Backoff LM

As described in Table 4, the LM text is collected from multiple sources. The top N most frequent words are selected as a vocabulary from the full-word text. For English, 150k most frequent words are used to generate modified Kneser-Ney smoothed 4-gram and 5-gram full-word LMs . Similarly for German, 150k and 200k full-word vocabularies are selected to generate 5-gram LMs.

3.3. Sub-lexical LMs

For an open vocabulary speech recognition, sub-lexical units are used in the language modeling for German LVCSR [11]. In general, a LM comprising sub-lexical units with or without a fraction of full-words is called a sub-lexical LM. In general, morphemes could be extracted using linguistic or data-driven morphological decomposition. When sub-lexical LMs are used, the data sparsity problem is relatively reduced compared to the full-word LMs, leading to lower OOV rates and higher lexical coverage. Furthermore, as the count based statistics are improved, the LM probability estimates are relatively better estimated compared to a full-word LM [10, 11, 12].

In this work, words are decomposed using a Morfessor [23]. Word decomposition model is trained using unique words that occur more than 5 times in the LM text. Low frequency words are excluded to avoid noise that are harmful during training. This model is also used to decompose new words. The decomposed words are processed so as to produce a cleaner set of sub-lexical units and to avoid very short units which are usually difficult to recognize. This is found to be helpful to improve the final WER. To generate sub-lexical LMs, 200k hybrid vocabulary is selected, where top-most 5k full-word forms are preserved. Standard N-gram backoff models are created using SRILM toolkit [24].

[3]http://www.iwslt2013.org/59.php

3.4. Maximum Entropy LMs

Alternatively, for German LVCSR, state-of-the-art MaxEnt LM is generated to capture the long range dependencies [25]. In principle, MaxEnt LM uses the information obtained from multiple knowledge sources as feature constraints. The knowledge sources could be different types of features having different constraints (i.e., probability distribution functions). MaxEnt LM estimates a unified model in a feature space by selecting the distribution function of the highest entropy satisfying all the constraints from an intersection of all the imposed feature constraints. If w is a word/morpheme taken from a vocabulary W, $f(.)$ is the feature function, λ is an optimal weight, h is the context, $Z(h)$ is the normalization factor for all the seen contexts, MaxEnt model can be computed using Eq. 1.

$$p_{me}(w|h) = \frac{e^{\sum_i \lambda_i f_i(w,h)}}{Z(h)} \tag{1}$$

$$\text{Where, } Z(h) = \sum_{w_i \epsilon W} e^{\sum_j \lambda_j f_j(w_i,h)}$$

3.5. Adaptation

In general, adapted LMs are known to perform better than non-adapted LMs in cases of domain mis-match or if the LM corpus is diverse. In this paper, the LM data is obtained from multiple domains for LVCSR. It is often unrealistic to significantly reduce the WER without adapting the LM to in-domain data [26]. For this reason, we apply LM adaptation over MaxEnt LMs. Here, Maximum a-posteriori (*MAP*) adaptation is performed, using Gaussian priors over the generated MaxEnt models (cf. Section 3.4). The MaxEnt model is trained on background data including the N-gram features of the in-domain data. The prior parameters computed from the background data are used to learn the parameters from the in-domain data. During MaxEnt training, the prior has zero mean during Gaussian prior smoothing. But during adaptation, the prior distribution is centered at the background data parameters. The regularized log-likelihood of the adaptation training data is maximized during adaptation.

As an in-domain data, two different types of adaptation, namely supervised and unsupervised are investigated [25]. In supervised adaptation, the development data is used as an in-domain data. Whereas, for an unsupervised adaptation, the automatic transcriptions are used from the first pass recognition. Here, the adaptation is performed over both morpheme and feature based MaxEnt models. The 5-gram MaxEnt and adapted models are created using SRILM-extension [27].

In general, N-gram backoff LMs are known to perform better in capturing the short range context dependencies.

When the data is sufficiently available, the likelihood estimates of the frequently occurring N-grams are generally better estimated and reliable. In this work, morphemic MaxEnt LMs are linearly interpolated with N-gram LMs [28].

3.6. Perplexity

Perplexity is a entropy related metric which measures the average branching factor for the LM, during search. On the other hand, perplexities across various systems can only be compared when the (same) finite vocabulary is used. The word level standard equation of the perplexity (PP_w) in log domain is :

$$PP_w(w_1^k) = log\left[\prod_{l=1}^{K} p(w_l|w_h)\right]^{-\frac{1}{K}} \quad (2)$$

Thus, Eq. 2 is renormalized using at character level as:

$$PP_c(w_1^k) = log\left[\prod_{l=1}^{K} p(w_l|w_h)\right]^{-\frac{1}{K}\frac{K}{K_c}} \quad (3)$$

Where, K is the total number of words observed in the recognition corpus. K_c represents the actual number of characters including word boundaries and a representative character per sentence-end token. Thus, using Eq. 3, full-word LM and the sub-lexical LM could be easily compared.

3.7. Lexical Modeling

The full-word lexicon consists of 150k words for English LVCSR. Similarly, lexicons consisting of 150k and 200k full-words are generated for German LVCSR. For most of the full-words as the pronunciations are not available, statistical grapheme-to-phoneme (G2P) conversion toolkit is used for both the languages [29]. The full-word pronunciations are aligned to its corresponding sequence of morphemic sub-lexical units using the expectation-maximization (EM) algorithm as described in [12]. Thereby, lexicon is generated using the sub-lexical entries of size 200k.

3.8. Word Reconstruction

For sub-lexical experiments, full-words are needed to be reconstructed from the morphemes. An identifier '+' is marked at the end of each non-boundary morpheme. After recognition, the recognized morphemes are combined using the predefined marker to regenerate the full-words. For example: *wasch+ masch+ ine* $\rightarrow$ *waschmaschine* (washing machine in English). Alternatively, the effective OOV rate of any corpus is computed in such a way that a word is considered an OOV if and only if it is not found in the vocabulary and it is not possible to compose it using in-vocabulary sub-lexical units.

4. Recognition Setup

The evaluation systems have a multi-pass recognition setup. In an initial non-adapted pass, a first transcription is obtained, which is used for the CMLLR-adapted recognition pass. The development and evaluation corpus statistics for both the languages are shown in Table 5.

Table 5: *Details of the IWSLT-13 Recognition Corpus*

Language	**Corpus**	**#Duration (hrs.)**
English	dev2012	2.0
	tst2011	1.3
	tst2012	2.2
	tst2013	4.8
German	dev2012	3.3
	tst2013	3.2

For the English LVCSR system, CMU segmentation is used [30]. A 4-gram domain adapted backoff LM is created to construct the search space and 5-gram LM is used for rescoring word lattices. For our alternative system (system 2), an non-adapted and a CMLLR-pass are performed as in system-1. In addition, a third recognition pass with MLLR adaptation is performed. Finally, the word lattices are rescored. Confusion network based system combination is used to combine the results of both systems.

For the German LVCSR system, two different systems are experimented with LIUM [31] and RWTH audio segmentation [32]. 5-gram domain adapted backoff LM is created to construct the search space. This recognition setup is similar to the system-1 of the English LVCSR. After the speaker adaptation, N-best (N=5000) lists are generated from the lattices for LM rescoring. The N-best lists are rescored using the interpolated LMs as described in Section 3.5. Similarly, the advantages of both the full-word and the sub-lexical systems are combined using confusion network decoding.

5. Results

In this Section, detailed results for the various systems are described in terms of the WER and the OOV rates. For both the languages, WERs for the development corpus are generated using the unofficial scoring script, where as the WERs for the evaluation corpus are obtained using official scoring script. For English LVCSR system, the recognition results are shown in Table 6. The WER of system-1 is better than system-2. Significant improvements are obtained using speaker adapted acoustic models over the speaker independent models. Further improvements are obtained using confusion network decoding. In addition, noticeable WERs are reported on the tst2011 and tst2012 corpora for IWSLT-2013 evaluation, compared to our previous IWSLT-2012 WERs for English LVCSR as shown in Table 7. Test transcriptions are not released by the IWSLT-13 evaluation committee, yet.

For the German LVCSR system, the first set of experiments are shown in Table 8. The BN features used in the evaluation system were optimized using the 200k sub-lexical LM, as it is better than the 150k or 200k full-word system interms of the WER. Thus, the recognition results using 150k vocabulary are not shown in this paper. The experiments were carried out with RWTH segmentation and sub-lexical language models containing 200k sub-lexical units. As can

Table 6: *WERs[%] of the English LVCSR system (OOV Rate:0.7, dev2012 PPL:129, Vocabulary size:150k).*

Corpus	Pass	System-1	System-2
dev2012	VTLN	17.6	21.9
	CMLLR	15.2	18.5
	MLLR	-	18.8
	LM-rescoring	14.8	17.9
	CN decoding	**14.4**	
tst2011		**10.2**	
tst2012		**11.3**	
tst2013		**16.0**	

Table 7: *Progressive WER [%] improvements : IWSLT-12 Vs. IWSLT-13 English LVCSR Systems*

Corpus	IWSLT 2012	IWSLT 2013	Rel. gain
tst2011	13.4	10.2	23.9
tst2012	13.6	11.3	20.3
tst2013	-	16.0	-

be seen in Table 8, the deep unilingual BN features trained on out-of-domain BN/BC data did not result in better WER compared to the shallow ones (1st and 3rd rows). Including multiple languages in the BN training improved the results significantly, and the performance gap increased further after the speaker adaptation step (3rd and 4th rows), similar to our observation in [8]. Furthermore, the results also show that the deep structure is more beneficial for multilingual training and outperforms the shallow multilingual BN (2nd, 4th rows). Different types of last hidden layers described in Subsection 2.2.1 were also investigated. Applying language dependent hidden layers between the bottleneck and output layer did not resulted in lower error rate (5th row). On the contrary, if the number of parameters were increased by larger language independent hidden layer further reduction in WER (6th row) is observed.

Table 8: *WER[%] comparison of speaker independent (SI) and speaker adapted (SA) uni- and multilingual BN features with different structures - German LVCSR with* **no** *word compounding (***Seg***: audio segmentation,* **SI***: speaker independent models,* **SA***: speaker adapted models)*

AM		Seg	Dev2012		Eval2013	
			SI	SA	SI	SA
BN features	Shallow	RWTH	22.3	20.1	30.0	27.5
	+multilingual		21.7	19.1	29.4	26.1
	Deep		22.1	20.5	30.1	28.1
	+multilingual		20.9	19.0	28.0	25.8
	+lang.dep.hidden		20.8	19.1	27.9	26.1
	+large hidden		**20.6**	**18.8**	**27.7**	**25.7**
		LIUM	20.8	19.0	27.9	25.9

Table 9: *Recognition results for 200k German LVCSR with* **no** *word compounding (***FW***: full-word system,* **Crp***: corpus,* **MW***: sub-lexical system, PP_w: word-level perplexity, PP_c: character level perplexity,* **unsp***: unsupervised adapted LM ,* **CN***: confusion network decoding,* **CER***: character error rate,* **Effective OOV rate** *:- Dev:0, eval:0.9)*

Expt.	Crp	LM	Adap	PP_w/PP_c	WER [%]	CER [%]
FW	dev	backoff	no	314/2.1	19.6	7.6
	eval			226/2.2	26.0	15.6
MW	dev	backoff	no	284/2.2	18.8	7.5
		+ME		282/2.2	18.8	7.5
	eval	backoff	no	240/2.3	25.4	15.4
		+ME+unsp	yes	239/2.3	25.4	15.4
CN dec. MW+FW	dev	backoff	no	–	**18.4**	7.5
	eval			–	**25.2**	15.4

Alternatively, as shown in Table 9, non-adapted and adapted MaxEnt models are applied on the morpheme systems interpolated with the backoff LM. Character-level perplexities are shown for fair comparison between full-word and morpheme based systems. Applying LM adaptation did not affect either the perplexity or the WER for both development and evaluation corpus. To capture the advantages of both the sub-lexical and full-word systems, system combination is used. Using confusion network decoding based system combination, further improvements are achieved compared to the stand-alone sub-lexical based system.

6. Conclusions

In this paper, the descriptions of the German and English LVCSR systems developed by the RWTH Aachen for the IWSLT 2013 evaluation are presented. Here, state-of-the-art acoustic level multilingual features, domain dependent language modeling, supervised and unsupervised adaptation and system combination of subsystems are experimented. Noticeable contribution of the improvements were achieved because of the use of multilingual features. Language model adaptation did not affect the WER. Although sub-lexical systems performed significantly better than the full-word systems, system combination outperformed all other systems. The RWTH produced competitive results for German and English LVCSRs in the IWSLT 2013 evaluation campaign.

7. Acknowledgements

This work was partly funded by the European Community's 7^{th} Framework Programme under the project EU-Bridge (FP7-287658) and partly realized under the Quaero Programme, funded by OSEO, French State agency for innovation. Hermann Ney was partially supported by a senior chair award from DIGITEO, a French research cluster in Ile-de-France.

8. References

[1] H. Hermansky, D. P. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, June 2000, pp. 1635 – 1638.

[2] F. Grézl, M. Karafiát, S. Kontár, and J. Černocký, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Honolulu, Hawaii, USA, Apr. 2007, pp. 757 – 760.

[3] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. Norwell, MA, USA: Kluwer Academic Publishers, 1993.

[4] A. Stolcke, F. Grézl, M.-Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006, pp. 321–324.

[5] C. Plahl, R. Schlüter, and H. Ney, "Cross-lingual Portability of Chinese and English Neural Network Features for French and German LVCSR," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, Hawaii, Dec. 2011, pp. 371 – 376.

[6] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the Use of a Multilingual Neural Network Front-End," in *Proc. of Interspeech*, Brisbane, Australia, Sept. 2008, pp. 2711–2714.

[7] K. Veselý, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *IEEE Workshop on Spoken Language Technology*, Miami, Florida, USA, Dec. 2012, pp. 336–341.

[8] Z. Tüske, R. Schlüter, and H. Ney, "Multilingual Hierarchical MRASTA Features for ASR," in *Interspeech*, Lyon, France, Aug. 2013, pp. 2222–2226.

[9] Z. Tüske, J. Pinto, D. Willett, and R. Schlüter, "Investigation on cross- and multilingual MLP features under matched and mismatched acoustical conditions," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013, pp. 7349–7353.

[10] M. Bisani and H. Ney, "Open Vocabulary Speech Recognition with Flat Hybrid Models," in *Interspeech*, Lisbon, Portugal, Sept. 2005, pp. 725 – 728.

[11] M. Shaik, A. El-Desoky, R. Schlüter, and H. Ney, "Hybrid Language Models Using Mixed Types of Sublexical Units for Open Vocabulary German LVCSR," in *Interspeech*, Florence, Italy, Aug. 2011, pp. 1441 – 1444.

[12] A. El-Desoky, M. Shaik, R. Schlüter, and H. Ney, "Sublexical language models for German LVCSR," in *IEEE Workshop on Spoken Language Technology*, Berkeley, CA, USA, Dec. 2010, pp. 159 – 164.

[13] M. Nußbaum-Thom, S. Wiesler, M. Sundermeyer, C. Plahl, S. Hahn, R. Schlüter, and H. Ney, "The RWTH 2009 quaero ASR evaluation system for English and German," in *Interspeech*, Makuhari, Chiba, Japan, Sept. 2010, pp. 1517 – 1520.

[14] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.

[15] H. Hermansky and P. Fousek, "Multi-resolution RASTA filtering for TANDEM-based ASR," in *Interspeech*, Lisbon, Portugal, Sept. 2005, pp. 361–364.

[16] F. Valente and H. Hermansky, "Hierarchical and parallel processing of modulation spectrum for ASR applications," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Las Vegas, Nevada, USA, Mar. 2008, pp. 4165–4168.

[17] C. Plahl, R. Schlüter, and H. Ney, "Hierarchical Bottle Neck Features for LVCSR," in *Interspeech*, Makuhari, japan, Sept. 2010, pp. 1197–1200.

[18] Z. Tüske, R. Schlüter, and H. Ney, "Deep hierarchical bottleneck MRASTA features for LVCSR," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013, pp. 6970–6974.

[19] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171 – 185, 1995.

[20] G. Stemmer, F. Brugnara, and D. Giuliani, "Adaptive training using simple target models," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Philadelphia, Pennsylvania, USA, Mar. 2005, pp. 997–1000.

[21] S. S. Chen and P. S. Gopalakrishnan, "Clustering via the bayesian information criterion with applications in speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1998, pp. 645–648.

[22] S. Peitz, S. Mansour, M. Freitag, M. Feng, M. Huck, J. Wuebker, M. Nuhn, M. Nußbaum-Thom, and H. Ney, "The RWTH Aachen Speech Recognition and Machine Translation System for IWSLT 2012," in *The International Workshop on Spoken Language Translation*, Hongkong, Dec. 2012, pp. 69–76.

[23] M. Creutz and K. Lagus, "Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0," Computer and Information Science Helsinki University of Technology, Finland, Tech. Rep., Mar. 2005.

[24] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. Int. Conf. on Spoken Language Processing*, vol. 2, Denver, Colorado, USA, Sept. 2002, pp. 901 – 904.

[25] M. Shaik, A. El-Desoky, R. Schlüter, and H. Ney, "Investigation of Maximum Entropy Hybrid Language Models for Open Vocabulary German and Polish LVCSR," in *Interspeech*, Portland, OR, USA, Sept. 2012.

[26] C. Chelba and A. Acero, "Adaptation of maximum entropy capitalizer: Little data can help a lot," *Computer Speech and Language*, vol. 20, no. 4, pp. 382 – 399, 2006.

[27] T. Alumäe and M. Kurimo, "Efficient Estimation of Maximum Entropy Language Models with N-gram features: an SRILM extension," in *Interspeech*, Chiba, Japan, September 2010.

[28] C. Samuelsson and W. Reichl, "A class-based language model for large-vocabulary speech recognition extracted from part-of-speech statistics," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, Phoenix, AZ, USA, Mar. 1999, pp. 537 – 540.

[29] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434 – 451, May 2008.

[30] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proc. DARPA Speech Recognition Workshop*, Chantilly, VA, USA, Feb. 1997, pp. 97–99.

[31] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, "An Open-source State-of-the-art Toolbox for Broadcast News Diarization," in *Interspeech*, Lyon, France, aug 2013, pp. 1477 – 1481.

[32] D. Rybach, C. Gollan, R. Schlüter, and H. Ney, "Audio Segmentation for Speech Recognition using Segment Features," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, Apr. 2009, pp. 4197 – 4200.

EU-BRIDGE MT: Text Translation of Talks in the EU-BRIDGE Project

[*]Markus Freitag, [*]Stephan Peitz, [*]Joern Wuebker, [*]Hermann Ney,
[‡]Nadir Durrani, [‡]Matthias Huck, [‡]Philipp Koehn,
[†]Thanh-Le Ha, [†]Jan Niehues, [†]Mohammed Mediani, [†]Teresa Herrmann, [†]Alex Waibel,
[§]Nicola Bertoldi, [§]Mauro Cettolo, [§]Marcello Federico

[*]RWTH Aachen University, Aachen, Germany
[‡]University of Edinburgh, Edinburgh, Scotland
[†]Karlsruhe Institute of Technology, Karlsruhe, Germany
[§]Fondazione Bruno Kessler, Trento, Italy

[*]{freitag,peitz,wuebker,ney}@cs.rwth-aachen.de
[‡]{ndurrani,mhuck,pkoehn}@inf.ed.ac.uk
[†]{thanh-le.ha,jan.niehues,teresa.herrmann,alex.waibel}@kit.edu
[†]mmediani@ira.uka.de
[§]{bertoldi,cettolo,federico}@fbk.eu

Abstract

EU-BRIDGE[1] is a European research project which is aimed at developing innovative speech translation technology. This paper describes one of the collaborative efforts within EU-BRIDGE to further advance the state of the art in machine translation between two European language pairs, English→French and German→English. Four research institutions involved in the EU-BRIDGE project combined their individual machine translation systems and participated with a joint setup in the machine translation track of the evaluation campaign at the 2013 International Workshop on Spoken Language Translation (IWSLT).

We present the methods and techniques to achieve high translation quality for text translation of talks which are applied at RWTH Aachen University, the University of Edinburgh, Karlsruhe Institute of Technology, and Fondazione Bruno Kessler. We then show how we have been able to considerably boost translation performance (as measured in terms of the metrics BLEU and TER) by means of system combination. The joint setups yield empirical gains of up to 1.4 points in BLEU and 2.8 points in TER on the IWSLT test sets compared to the best single systems.

1. Introduction

The International Workshop on Spoken Language Translation [1] hosts a yearly open evaluation campaign on the translation of TED talks [2]. The TED talks task is challenging from the perspective of automatic speech recognition (ASR) and machine translation (MT) as it involves spontaneous speech and heterogeneous topics and styles. The task is open domain, with a wide range of heavily dissimilar subjects and jargons across talks. IWSLT subdivides the task and separately evaluates *automatic transcription of talks from audio to text*, *speech translation of talks from audio*, and *text translation of talks* as three different tracks [3, 4]. The training data is constrained to the corpora specified by the organizers. The supplied list of corpora comprises a large amount of publicly available monolingual and parallel training data, though, including WIT[3] [5], Europarl [6], MultiUN [7], the English and French Gigaword corpora as provided by the Linguistic Data Consortium [8], and the News Crawl, 10^9 and News Commentary corpora from the WMT shared task training data [9]. For the two "official" language pairs [1] for translation at IWSLT 2013, English→French and German→English, these resources allow for building of systems with state-of-the-art performance by participants.

The EU-BRIDGE project is funded by the European Union under the Seventh Framework Programme (FP7) [10] and brings together several project partners who have each previously been very successful in contributing to advancements in automatic speech recognition and statistical machine translation. A number of languages and language pairs (both well-covered and under-resourced ones) are tackled with ASR and MT technology with different use cases in mind. Four of the EU-BRIDGE project partners are particularly experienced in machine translation for European language pairs: RWTH Aachen University (RWTH), the University of Edinburgh (UEDIN), Karlsruhe Institute of Technology (KIT), and Fondazione Bruno Kessler (FBK) have all regularly participated in large-scale evaluation campaigns like IWSLT and WMT in recent years, thereby demonstrating their ability to continuously enhance their systems and promoting progress in machine translation. Machine trans-

[1] http://www.eu-bridge.eu

lation research within EU-BRIDGE has a strong focus on translation of spoken language. The IWSLT TED talks task constitutes an interesting framework for empirical testing of some of the systems for spoken language translation which are developed as part of the project.

The work described here is an attempt to attain translation quality beyond strong single system performance via system combination [11]. Similar cooperative approaches based on system combination have proven to be valuable for machine translation in other projects, e.g. in the Quaero programme [12, 13]. Within EU-BRIDGE, we built combined system setups for text translation of talks from English to French as well as from German to English. We found that the combined translation engines of RWTH, UEDIN, KIT, and FBK systems are very effective. In the rest of the paper we will give some insight into the technology behind the combined engines which have been used to produce the joint EU-BRIDGE submission to the IWSLT 2013 MT track.

The remainder of the paper is structured as follows: We first describe the individual English→French and German→English systems by RWTH Aachen University (Section 2), the University of Edinburgh (Section 3), Karlsruhe Institute of Technology (Section 4), and Fondazione Bruno Kessler (Section 5), respectively. We then present the techniques for machine translation system combination which have been employed to obtain consensus translations from the outputs of the individual systems of the project partners (Section 6). Experimental results in BLEU [14] and TER [15] are given in Section 7. A brief error analysis on selected examples from the test data has been conducted which we discuss in Section 8. We finally conclude the paper with Section 9.

2. RWTH Aachen University

RWTH applied both the phrase-based (*RWTH scss*) and the hierarchical (*RWTH hiero*) decoder implemented in RWTH's publicly available translation toolkit Jane [16, 17, 18, 19]. The model weights of all systems were tuned with standard Minimum Error Rate Training [20] on the provided dev2010 set. RWTH used BLEU as optimization objective. Language models were created with the SRILM toolkit [21]. All RWTH systems include the standard set of models provided by Jane.

For English→French, the final setups for *RWTH scss* and *RWTH hiero* differ in the amount of training data and in the choice of models.

For the English→French hierarchical setup the bilingual data was limited to the in-domain WIT^3 data, News Commentary, Europarl, and Common Crawl corpora. The word alignment was created with *fast_align* [22]. A language model was trained on the target side of all available bilingual data plus $\frac{1}{2}$ of the Shuffled News corpus and $\frac{1}{4}$ of the French Gigaword Second Edition corpus. The monolingual data selection for using only parts of the corpora is based on cross-entropy difference as described in [23]. The hierarchical system was extended with a second translation model. The additional translation model was trained on the WIT^3 portion of the training data only.

For the English→French phrase-based setup, RWTH utilized all available parallel data and trained a word alignment with GIZA++ [24]. The same language model as in the hierarchical setup was used. RWTH applied the following supplementary features for the phrase-based system: a lexicalized reordering model [25], a discriminative word lexicon [26], a 7-gram word class language model [27], a continuous space language model [28], and a second translation model from the WIT^3 portion of the training data only.

For German→English, RWTH decompounded the German source in a preprocessing step [29] and applied part-of-speech-based long-range verb reordering rules [30]. Both systems *RWTH scss* and *RWTH hiero* rest upon all available bilingual data and word alignment obtained with GIZA++. A language model was trained on the target side of all available bilingual data plus $\frac{1}{2}$ of the Shuffled News corpus and $\frac{1}{4}$ of the English Gigaword v3 corpus, resulting in a total of 1.7 billion running words.

In both German→English systems, RWTH applied a more sophisticated discriminative phrase training method. Similar to [31], a gradient-based method is used to optimize a maximum expected BLEU objective, for which we define BLEU on the sentence level with smoothed 3-gram and 4-gram precisions. RWTH performed discriminative training on the WIT^3 portion of the training data.

The German→English phrase-based system was furthermore improved by a lexicalized reordering model and 7-gram word class language model. RWTH finally applied domain adaptation by adding a second translation model to the decoder which was trained on the WIT^3 portion of the data only. This second translation model was likewise improved with discriminative phrase training.

3. University of Edinburgh

UEDIN's systems were trained using the Moses system [32], replicating the settings described in [33] developed for the 2013 Workshop on Statistical Machine Translation. The characteristics of the system include: a maximum sentence length of 80, grow-diag-final-and symmetrization of GIZA++ alignments, an interpolated Kneser-Ney smoothed 5-gram language model with KenLM [34] used at runtime, a lexically-driven 5-gram operation sequence model [35] with four additional supportive features (two gap-based penalties, one distance-based feature and one deletion penalty), msd-bidirectional-fe lexicalized reordering, sparse lexical and domain features [36], a distortion limit of 6, 100-best translation options, minimum Bayes risk decoding [37], cube pruning [38] with a stack size of 1000 during tuning and 5000 during testing and the no-reordering-over-punctuation heuristic. UEDIN used the compact phrase table representation by [39]. For English→German, UEDIN used a sequence model over morphological tags.

The UEDIN systems were tuned on the dev2010 set made available for the IWSLT 2013 workshop. Tuning was performed using the *k*-best batch MIRA algorithm [40] with a maximum number of iterations of 25. BLEU was used as the metric to evaluate results.

While UEDIN's main submission also includes sequence models and operation sequence models over Brown word clusters, these setups were not finished in time for the contribution to the EU-BRIDGE system combination.

4. Karlsruhe Institute of Technology

The KIT translations have been generated by an in-house phrase-based translations system [41]. The models were trained on the Europarl, News Commentary, WIT^3, Common Crawl corpora for both directions and WMT 10^9 for English→French and the additional monolingual training data. The big noisy 10^9 and Crawl corpora were filtered using an SVM classifier [42]. In addition to the standard preprocessing, KIT used compound splitting [29] for the German text.

In both translation directions, KIT performed reordering using two models. KIT encoded different reorderings of the source sentences in a word lattice. For the English→French system, only short-range rules [43] were used to generate these lattices. For German→English, long-range rules [44] and tree-based reordering rules [45] were used as well. The part-of-speech (POS) tags needed for these rules were generated by the TreeTagger [46] and the parse trees by the Stanford Parser [47]. In addition, KIT scored the different reorderings of both language pairs using a lexicalized reordering model [48].

The phrase tables of the systems were trained using GIZA++ alignment for the English→French task and using a discriminative alignment [49] for the German→English task. KIT adapted the phrase table to the TED domain using the back off approach and by also adapting the candidate selection [50]. In addition to the phrase table probabilities, KIT modeled the translation process by a bilingual language model [51] and a discriminative word lexicon [52]. For the German→English task, a discriminative word lexicon with source and target context features was applied, while only the source context features were employed for the English→French task.

During decoding, KIT used several language models to adapt the system to the task and to better model the sentence structure by means of class-based *n*-grams. For the German→English task, KIT used one language model trained on all data, an in-domain language model trained only on the WIT^3 corpus and one language model trained on 5 M sentences selected using cross-entropy difference [23]. Furthermore, KIT used an RBM-based language model [53] trained on the WIT^3 corpus. Finally, KIT also used a class-based language model, trained on the WIT^3 corpus using the MKCLS [54] algorithm to cluster the words. For the English→French translation task, KIT linearly combined the language models trained on WIT^3, Europarl, News Commentary, 10^9, and Common Crawl by minimizing the perplexity on the development data. For the class-based language model, KIT utilized in-domain WIT^3 data with 4-grams and 50 clusters. In addition, a 9-gram POS-based language model derived from LIA POS tags [55] on all monolingual data was applied.

KIT optimized the log-linear combination of all these models on the provided development data using Minimum Error Rate Training [20].

5. Fondazione Bruno Kessler

The FBK component of the system combination corresponds to the "contrastive 1" system of the official FBK submission. The FBK system was built upon a standard phrase-based system using the Moses toolkit [32], and exploited the huge amount of parallel English→French and monolingual French training data, provided by the organizers. It featured a statistical log-linear model including a filled-up phrase translation model [56] and lexicalized reordering models (RMs), two French language models (LMs), as well as distortion, word, and phrase penalties. In order to focus it on TED specific domain and genre, and to reduce the size of the system, data selection by means of IRSTLM toolkit [57] was performed on the whole parallel English→French corpus, using the WIT^3 training data as in-domain data. Different amount of data are selected from each available corpora but the WIT^3 data, for a total of 66 M English running words. Two TMs and two RMs were trained on WIT^3 and selected data, separately, and combined using the fill-up (for TM) and back-off (for RM) techniques, using WIT^3 as primary component. The French side of WIT^3 and selected data were employed to estimate a mixture language model [58]. A second huge French LM was estimated on the monolingual French available data of about 2.4 G running words. Both LMs have order five and were smoothed by means of the interpolated Improved Kneser-Ney method [59]; the second LM was also pruned-out of singleton *n*-gram ($n > 2$). Tuning of the system was performed on dev2010 by optimizing BLEU using Minimum Error Rate Training [20]. It is worth noticing that the dev2010 and test2010 data were added to the training data in order to build the system actually employed in the translation of test2011, test2012, test2013.

6. System Combination

System combination is used to produce consensus translations from multiple hypotheses which are outputs of different translation engines. The consensus translations can be better in terms of translation quality than any of the individual hypotheses. To combine the engines of the project partners for the EU-BRIDGE joint setups, we applied a system combination implementation that has been developed at RWTH Aachen University.

The basic concept of RWTH's approach to machine translation system combination has been described by Matusov et al. [60]. This approach includes an enhanced alignment and reordering framework. Alignments between the system outputs are learned using METEOR [61]. A confusion network is then built using one of the hypotheses as "primary" hypothesis. We do not make a hard decision on which of the hypotheses to use for that, but instead combine all possible confusion networks into a single lattice. Majority voting on the generated lattice is performed using the prior probabilities for each system as well as other statistical models, e.g. a special *n*-gram language model which is learned on the input hypotheses. Scaling factors of the models are optimized using the Minimum Error Rate Training algorithm. The translation with the best total score within the lattice is selected as consensus translation.

7. Results

In this section, we present our experimental results on the two translation tasks, German→English and English→French.

7.1. German→English

RWTH Aachen University, the University of Edinburgh, and Karlsruhe Institute of Technology participated in the German→English translation task. The individual results as well as the system combination results are given in Table 1. RWTH's phrase-based translation (*scss*) is the best of the four single systems on test2010. The pairwise difference of the single system performance is up to 1.5 points in BLEU. In the end each system was needed to reach the performance of our final system combination submission. We optimized our system combination parameters on test2010. With the standard set of features, we got a gain of 1.5 BLEU on dev2010 and 1.2 BLEU on test2010 compared to the best single system. We tried different setups; also one which includes the large language model from RWTH's single systems as additional language model (+ *bigLM*). The translation quality in terms of BLEU improves by 0.2 on test2010 but degrades by 0.4 on dev2010. The TER scores were improved on both test sets, though. We decided to submit the system combination including *bigLM* as primary submission and the system combination without the large language model as secondary submission.

7.2. English→French

RWTH Aachen University, the University of Edinburgh, Karlsruhe Institute of Technology, and Fondazione Bruno Kessler participated in the English→French translation task. In Table 2 the results of the individual systems and our best system combination results are listed. The best individual system was provided by UEDIN. In this language pair the pairwise difference of the single systems was up to 1.5 points in BLEU. As in the German→English translation task, we tried to optimize our parameters on test2010. We got a large improvement on test2010 of 2.1 points in BLEU, but got only a slight improvement of 0.3 BLEU on dev2010. After changing the optimization set to dev2010, we got comparable improvements on both test sets. On dev2010 we got an improvement of 1.4 points in BLEU and on test2010 an improvement of 0.8 points in BLEU. On both test sets the performance in TER was similar or even better compared to the system combination optimized on test2010. We decided to submit the system combination optimized on dev2010 as primary submission.

Table 1: Results for the German→English translation task. Bold font indicates system combination results that are significantly better than the best single system ($p < 0.05$).

system	dev2010		test2010	
	BLEU	TER	BLEU	TER
RWTH scss	33.3	47.0	31.4	49.3
KIT	33.6	46.5	31.1	49.5
RWTH hiero	33.0	46.8	30.7	49.5
UEDIN	32.1	47.3	29.9	49.6
sc	**34.8**	**44.9**	**32.6**	**47.4**
sc + bigLM	**34.4**	**44.4**	**32.8**	**46.5**

Table 2: Results for the English→French translation task. Bold font indicates system combination results that are significantly better than the best single system with $p < 0.05$. Italic font indicates system combination results that are significantly better than the best single system with $p < 0.1$.

system	dev2010		test2010	
	BLEU	TER	BLEU	TER
UEDIN	29.4	55.4	33.2	49.8
RWTH scss	28.8	55.4	32.8	49.2
KIT	28.8	55.7	32.6	49.3
FBK	27.5	57.0	32.1	50.0
RWTH hiero	28.0	56.3	31.7	49.9
sc opt dev10	*30.8*	**53.8**	*34.0*	**48.1**
sc opt test10	29.7	55.2	**35.3**	**48.2**

8. Error Analysis

We carried out a restricted manual error analysis to compare the outputs of each single system to the final system combination output for some example sentences. In Figure 1 and Figure 2 the TER scores of all translations of two selected sentences from the German→English translation direction are given. In both sentences system combination outperforms each single system.

KIT (TER Score: 60.00 (9.0/ 15.0))	
hyp	except for your contribution , whatever it may be .
shifted hyp	——— – —- except for your —— contribution it , whatever —- may be .
edited hyp	——— – —- except for your —— contribution it , whatever —- may be .
ref	continue to show up for your piece of it , whatever that might be .
RWTH hiero (TER Score: 66.67 (10.0/ 15.0))	
hyp	is still there for the post , whatever it may be .
shifted hyp	——— is still there for —- the post it , whatever —- may be .
edited hyp	——— is still there for —- the post it , whatever —- may be .
ref	continue to show up for your piece of it , whatever that might be .
RWTH scss (TER Score: 66.67 (10.0/ 15.0))	
hyp	for your contribution is still there , whatever it may be .
shifted hyp	——— – —- – for your contribution is still there , whatever it may be .
edited hyp	——— – —- – for your contribution is still there , whatever it may be .
ref	continue to show up for your ———— piece of it , whatever that might be .
UEDIN (TER Score: 66.67 (10.0/ 15.0))	
hyp	continue to be there for your contribution , which may be his time .
shifted hyp	continue to —- there for your contribution which may , be his time be .
edited hyp	continue to —- there for your contribution which may , be his time be .
ref	continue to show up for your piece of it , whatever that might be .
system combination (TER Score: 46.67 (7.0/ 15.0))	
hyp	continue to be there for your contribution , whatever it may be .
shifted hyp	continue to be there for your —— contribution it , whatever —- may be .
edited hyp	continue to be there for your —— contribution it , whatever —- may be .
ref	continue to show up for your piece of it , whatever that might be .

Figure 1: Error analysis for sentence 715 (dev2010) in the German→English translation task.

KIT (TER Score: 52.63 (10.0/ 19.0))	
hyp	they can only be in conjunction with a number of other chemicals taken the mao advised .
shifted hyp	they can only be —— ——— – taken in conjunction with a other number of chemicals the mao advised .
edited hyp	they can only be —— ——— – taken in conjunction with a other number of chemicals the mao advised .
ref	they can only be taken orally if taken in conjunction with some other chemical that denatures the mao ———- .
RWTH hiero (TER Score: 47.37 (9.0/ 19.0))	
hyp	they can only be taken in combination with other chemicals , who turned the mao .
shifted hyp	they can only be —— ——— – taken in combination with chemicals other , who turned the mao .
edited hyp	they can only be —— ——— – taken in combination with chemicals other , who turned the mao .
ref	they can only be taken orally if taken in conjunction with some other chemical that denatures the mao .
RWTH scss (TER Score: 47.37 (9.0/ 19.0))	
hyp	they can only be consumed in connection with some other chemicals , which turned the mao .
shifted hyp	they can only be —— ——— – consumed in connection with some other chemicals , which turned the mao .
edited hyp	they can only be —— ——— – consumed in connection with some other chemicals , which turned the mao .
ref	they can only be taken orally if taken in conjunction with some other ———— chemical that denatures the mao .
UEDIN (TER Score: 36.84 (7.0/ 19.0))	
hyp	they can be used only in conjunction with other chemicals taken orally that denaturieren the mao .
shifted hyp	they can only be taken orally – used in conjunction with —- other chemicals that denaturieren the mao .
edited hyp	they can only be taken orally – used in conjunction with —- other chemicals that denaturieren the mao .
ref	they can only be taken orally if taken in conjunction with some other chemical that denatures the mao .
system combination (TER Score: 26.32 (5.0/ 19.0))	
hyp	they can only be taken in conjunction with other chemicals taken orally that turned the mao .
shifted hyp	they can only be taken orally – taken in conjunction with —- other chemicals that turned the mao .
edited hyp	they can only be taken orally – taken in conjunction with —- other chemicals that turned the mao .
ref	they can only be taken orally if taken in conjunction with some other chemical that denatures the mao .

Figure 2: Error analysis for sentence 90 (dev2010) in the German→English translation task.

Words marked with red are substitutions, blue are insertions, green are deletions and yellow are shifts. In Figure 1 the final system combination translation is build out of the beginning part of UEDIN and the end part of all other single systems. Combined, this new translation improves over all single systems in terms of TER. In Figure 2 the system combination output is basically a fixed version of the UEDIN translation. This results in a better TER score which needs two less edits.

9. Conclusion

For our participation in the MT track of the IWSLT 2013 evaluation campaign, four partners from the EU-BRIDGE project (RWTH Aachen University, University of Edinburgh, Karlsruhe Institute of Technology, Fondazione Bruno Kessler) provided a joint submission. Our combined EU-BRIDGE system setup for text translation of talks is part of our efforts within the project to deliver high-quality machine translation of spoken language.

By joining the outputs of the partners' different individual machine translation engines via a system combination framework we have been able to achieve significantly better translation performance (up to +1.4 BLEU and -2.8 TER). While each of the individual engines provides performance that is state-of-the-art for single systems, our results suggest that system combination techniques are still a fertile approach to benefit from diversity in collaborative efforts and thus progress towards even better quality.

In future research we intend to both improve single systems and to investigate novel methods and models in machine translation system combination for large-scale and real-world settings.

10. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n^{o} 287658.

11. References

[1] International Workshop on Spoken Language Translation 2013, http://www.iwslt2013.org.

[2] TED Talks, http://www.ted.com/talks.

[3] M. Federico, L. Bentivogli, M. Paul, and S. Stueker, "Overview of the IWSLT 2011 Evaluation Campaign," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, USA, Dec. 2011.

[4] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 Evaluation Campaign," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, December 2012.

[5] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[6] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proc. of the MT Summit X*, Phuket, Thailand, Sept. 2005.

[7] A. Eisele and Y. Chen, "MultiUN: A Multilingual Corpus from United Nation Documents," in *Proceedings of the Seventh conference on International Language Resources and Evaluation*, May 2010, pp. 2868–2872.

[8] Linguistic Data Consortium (LDC), http://www.ldc.upenn.edu.

[9] Shared Translation Task of the ACL 2013 Eighth Workshop on Statistical Machine Translation, http://www.statmt.org/wmt13/translation-task.html.

[10] European Commission Community Research and Development Infomation Service (CORDIS), "Seventh Framework Programme (FP7)," http://cordis.europa.eu/fp7/.

[11] E. Matusov, N. Ueffing, and H. Ney, "Computing Consensus Translation from Multiple Machine Translation Systems Using Enhanced Hypotheses Alignment," in *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2006, pp. 33–40.

[12] M. Freitag, S. Peitz, M. Huck, H. Ney, T. Herrmann, J. Niehues, A. Waibel, A. Allauzen, G. Adda, B. Buschbeck, J. M. Crego, and J. Senellart, "Joint WMT 2012 Submission of the QUAERO Project," in *NAACL 2012 Seventh Workshop on Statistical Machine Translation*, Montréal, Canada, June 2012, pp. 322–329.

[13] S. Peitz, S. Mansour, M. Huck, M. Freitag, H. Ney, E. Cho, T. Herrmann, M. Mediani, J. Niehues, A. Waibel, A. Allauzen, Q. K. Do, B. Buschbeck, and T. Wandmacher, "Joint WMT 2013 Submission of the QUAERO Project," in *ACL 2013 Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, Aug. 2013.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, USA, July 2002, pp. 311–318.

[15] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, Cambridge, MA, USA, Aug. 2006, pp. 223–231.

[16] D. Vilar, D. Stein, M. Huck, and H. Ney, "Jane: Open Source Hierarchical Translation, Extended with Reordering and Lexicon Models," in *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden, July 2010, pp. 262–270.

[17] Vilar, David and Stein, Daniel and Huck, Matthias and Ney, Hermann, "Jane: an advanced freely available hierarchical machine translation toolkit," *Machine Translation*, vol. 26, no. 3, pp. 197–216, Sept. 2012.

[18] M. Huck, J.-T. Peter, M. Freitag, S. Peitz, and H. Ney, "Hierarchical Phrase-Based Translation with Jane 2," *The Prague Bulletin of Mathematical Linguistics (PBML)*, vol. 98, pp. 37–50, Oct. 2012.

[19] J. Wuebker, M. Huck, S. Peitz, M. Nuhn, M. Freitag, J.-T. Peter, S. Mansour, and H. Ney, "Jane 2: Open

Source Phrase-based and Hierarchical Statistical Machine Translation," in *COLING '12: The 24th Int. Conf. on Computational Linguistics*, Mumbai, India, Dec. 2012, pp. 483–491.

[20] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003, pp. 160–167.

[21] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *Proc. of the Int. Conf. on Speech and Language Processing (ICSLP)*, vol. 2, Denver, CO, USA, Sept. 2002, pp. 901–904.

[22] C. Dyer, V. Chahuneau, and N. A. Smith, "A Simple, Fast, and Effective Reparameterization of IBM Model 2," in *Proceedings of NAACL-HLT*, Atlanta, GA, USA, June 2013, pp. 644–648.

[23] R. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *ACL (Short Papers)*, Uppsala, Sweden, July 2010, pp. 220–224.

[24] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[25] M. Galley and C. D. Manning, "A Simple and Effective Hierarchical Phrase Reordering Model," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '08, Honolulu, HI, USA, 2008, pp. 848–856.

[26] A. Mauser, S. Hasan, and H. Ney, "Extending statistical machine translation with discriminative and trigger-based lexicon models," in *Conference on Empirical Methods in Natural Language Processing*, Singapore, Aug. 2009, pp. 210–217.

[27] J. Wuebker, S. Peitz, F. Rietig, and H. Ney, "Improving Statistical Machine Translation with Word Class Models," in *Conference on Empirical Methods in Natural Language Processing*, Seattle, WA, USA, Oct. 2013, pp. 1377–1381.

[28] H. Schwenk, A. Rousseau, and M. Attik, "Large, Pruned or Continuous Space Language Models on a GPU for Statistical Machine Translation," in *NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, Montréal, Canada, June 2012, pp. 11–19.

[29] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *Proc. 10th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, Budapest, Hungary, Apr. 2003, pp. 347–354.

[30] M. Popović and H. Ney, "POS-based Word Reorderings for Statistical Machine Translation," in *International Conference on Language Resources and Evaluation*, 2006, pp. 1278–1283.

[31] X. He and L. Deng, "Maximum Expected BLEU Training of Phrase and Lexicon Translation Models," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, Jeju, Republic of Korea, Jul 2012, pp. 292–301.

[32] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *ACL 2007 Demonstrations*, Prague, Czech Republic, 2007.

[33] N. Durrani, B. Haddow, K. Heafield, and P. Koehn, "Edinburgh's Machine Translation Systems for European Language Pairs," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August 2013.

[34] K. Heafield, "KenLM: Faster and Smaller Language Model Queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, UK, July 2011, pp. 187–197.

[35] N. Durrani, H. Schmid, and A. Fraser, "A Joint Sequence Translation Model with Integrated Reordering," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, OR, USA, June 2011, pp. 1045–1054.

[36] E. Hasler, B. Haddow, and P. Koehn, "Sparse Lexicalised Features and Topic Adaptation for SMT," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, Dec. 2012, pp. 268–275.

[37] S. Kumar and W. Byrne, "Minimum Bayes-Risk Decoding for Statistical Machine Translation," in *Proc. Human Language Technology Conf. / North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL)*, Boston, MA, USA, May 2004, pp. 169–176.

[38] L. Huang and D. Chiang, "Forest Rescoring: Faster Decoding with Integrated Language Models," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, June 2007, pp. 144–151.

[39] M. Junczys-Dowmunt, "Phrasal Rank-Encoding: Exploiting Phrase Redundancy and Translational Relations for Phrase Table Compression," *The Prague Bulletin of Mathematical Linguistics*, vol. 98, pp. 63–74, 2012.

[40] C. Cherry and G. Foster, "Batch Tuning Strategies for Statistical Machine Translation," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Montréal, Canada, June 2012, pp. 427–436.

[41] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *International Conference on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

[42] M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel, "The KIT English-French Translation systems for IWSLT 2011," in *Proceedings of the eight International Workshop on Spoken Language Translation (IWSLT)*, 2011.

[43] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Skövde, Sweden, 2007.

[44] J. Niehues and M. Kolss, "A POS-Based Model for Long-Range Reorderings in SMT," in *Fourth Workshop on Statistical Machine Translation (WMT 2009)*, Athens, Greece, 2009.

[45] T. Herrmann, J. Niehues, and A. Waibel, "Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation," in *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*, Altanta, GA, USA, June 2013.

[46] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *International Conference on New Methods in Language Processing*, Manchester, United Kingdom, 1994.

[47] A. N. Rafferty and C. D. Manning, "Parsing three german treebanks: lexicalized and unlexicalized baselines," in *Proceedings of the Workshop on Parsing German*, 2008.

[48] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Pittsburgh, PA, USA, 2005.

[49] J. Niehues and S. Vogel, "Discriminative Word Alignment via Alignment Matrix Modeling." in *Proc. of Third ACL Workshop on Statistical Machine Translation*, Columbus, USA, 2008.

[50] J. Niehues and A. Waibel, "Detailed Analysis of different Strategies for Phrase Table Adaptation in SMT," in *Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, San Diego, CA, USA, Oct. / Nov. 2012.

[51] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, 2011.

[52] J. Niehues and A. Waibel, "An MT Error-Driven Discriminative Word Lexicon using Sentence Structure Features," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, Aug. 2013, pp. 512–520.

[53] ——, "Continuous Space Language Models using Restricted Boltzmann Machines," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, Dec. 2012.

[54] F. J. Och, "An Efficient Method for Determining Bilingual Word Classes." in *EACL'99*, 1999.

[55] F. Béchet, "LIA PHON: Un Systeme Complet de Phonétisation de Textes," *Traitement automatique des langues*, vol. 42, no. 1, pp. 47–67, 2001.

[56] A. Bisazza, N. Ruiz, and M. Federico, "Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, USA, Dec. 2011, pp. 136–143.

[57] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an open source toolkit for handling large scale language models," in *Interspeech*, 2008, pp. 1618–1621.

[58] M. Federico and R. De Mori, "Language modelling," *Spoken Dialogues with Computers*, pp. 199–230, 1998.

[59] F. James, "Modified Kneser-Ney Smoothing of n-gram Models," RIACS, Tech. Rep. 00.07, Oct. 2000.

[60] E. Matusov, G. Leusch, R. Banchs, N. Bertoldi, D. Dechelotte, M. Federico, M. Kolss, Y.-S. Lee, J. Marino, M. Paulik, S. Roukos, H. Schwenk, and H. Ney, "System Combination for Machine Translation of Spoken and Written Language," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 7, pp. 1222–1237, 2008.

[61] S. Banerjee and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments," in *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, Ann Arbor, MI, USA, June 2005, pp. 65–72.

The MIT-LL/AFRL IWSLT-2013 MT System[†]

Michaeel Kazi, Michael Coury, Elizabeth Salesky, Jessica Ray, Wade Shen, Terry Gleason

MIT/Lincoln Laboratory
Human Language Technology Group
244 Wood Street
Lexington, MA 02420, USA
{michaeel.kazi, michael.coury, elizabeth.salesky, jessica.ray, swade, tpg}@ll.mit.edu

Tim Anderson, Grant Erdmann, Lane Schwartz, Brian Ore, Raymond Slyh, Jeremy Gwinnup, Katherine Young, Michael Hutt

Air Force Research Laboratory
Human Effectiveness Directorate
2255 H Street
Wright-Patterson AFB, OH 45433
{timothy.anderson.20, grant.erdmann, lane.schwartz, brian.ore.ctr, raymond.slyh, jeremy.gwinnup.ctr, katherine.young.1.ctr, michael.hutt.ctr}@us.af.mil

Abstract

This paper describes the MIT-LL/AFRL statistical MT system and the improvements that were developed during the IWSLT 2013 evaluation campaign [1]. As part of these efforts, we experimented with a number of extensions to the standard phrase-based model that improve performance on the Russian to English, Chinese to English, Arabic to English, and English to French TED-talk translation task. We also applied our existing ASR system to the TED-talk lecture ASR task.

We discuss the architecture of the MIT-LL/AFRL MT system, improvements over our 2012 system, and experiments we ran during the IWSLT-2013 evaluation. Specifically, we focus on 1) cross-entropy filtering of MT training data, and 2) improved optimization techniques, 3) language modeling, and 4) approximation of out-of-vocabulary words.

1. Introduction

During the evaluation campaign for the 2013 International Workshop on Spoken Language Translation (IWSLT-2013) [1] our experimental efforts centered on 1) cross-entropy filtering of MT training data, and 2) improved optimization techniques, 3) language modeling, and 4) approximation of out-of-vocabulary words.

In this paper we describe improvements over our 2012 baseline systems and methods we used to combine outputs from multiple systems. For a more in-depth description of the 2012 baseline system, refer to [3].

The remainder of this paper is structured as follows. Section 2 presents our work on the MT task, and discusses language independent algorithms. Section 3 discusses our MT algorithms for specific language pairs. Section 4 describes final systems and results. Section 5 presents our work on the automatic speech recognition (ASR) task.

[†]This work is sponsored by the Air Force Research Laboratory under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

2. Machine Translation

2.1. IWSLT-2013 Data Usage

We submitted systems for the English-to-French, Russian-to-English, Chinese-to-English, Arabic-to-English, and Farsi-to-English MT tasks. We used data supplied by the evaluation for each language pair [2] for training the baseline system, and approved out-of-domain data for the remainder. Unless otherwise noted, we used the optimization data set `dev2010` supplied by IWSLT 2013.

2.2. Baseline MT System

Our baseline system implements a fairly standard SMT architecture allowing for training of a variety of word alignment types and rescoring models. It has been applied successfully to a number of different translation tasks in prior work, including prior IWSLT evaluations. The training/decoding procedure for our system is outlined in Table 1. Details of the training procedure are described in [4].

Training Process
1. Segment training corpus
2. Compute GIZA++, Berkeley and Competitive Linking Alignments (CLA) for segmented data [6] [11] [12]
3. Extract phrases for all variants of the training corpus
4. Split word-segmented phrases into characters
5. Combine phrase counts and normalize
6. Train language models from the training corpus
7. Train TrueCase models
8. Train source language repunctuation models
Decoding/Rescoring Process
1. Decode input sentences using base models
2. Add rescoring features (e.g. IBM model-1 score, etc.)
3. Merge n-best lists (if input is ASR n-best)
4. Rerank n-best list entries

Table 1: *Training/decoding process*

2.2.1. Phrase Table Training

When building our phrase table, we applied Kneser-Ney discounting [5] to the forward and backward translation probabilities of the

phrases extracted during word alignment. In the past, we have combined multiple word alignment strategies, as described in [6]. For the experiments described here, we used only IBM model 4 or 5 for word alignment (see [7] and [8]), to keep the statistics appropriate for discounting.

2.2.2. Baseline Language Model Training

During the training process we built n-gram language models (LMs) for use in decoding/rescoring. We performed TrueCasing and repunctuation using 2-gram language models and `disambig` from the SRI toolkit. The MIT Language Modeling Toolkit [10] was used to create interpolated Kneser-Ney LMs in all cases. Additional class-based language models were also trained using `mkcls` [9] for rescoring. Most systems made use of 7-gram language models for rescoring trained on the target side of the parallel text.

2.2.3. Optimization, Decoding, and Rescoring

Our translation model assumes a log-linear combination of phrase translation models, language models, etc.

$$\log P(\mathbf{E}|\mathbf{F}) \propto \sum_{\forall r} \lambda_r h_r(\mathbf{E}, \mathbf{F})$$

To optimize system performance we train scaling factors, λ_r, for both decoding and rescoring features so as to minimize an objective error criterion. In our baseline systems, this is done using a standard Powell-like grid search performed on a development set [13].

A full list of the independent model parameters that we used in our baseline system is shown in Table 2. All systems generated n-best lists that are then rescored and reranked using either a maximum likelihood (ML) or an minimum Bayes risk (MBR) criterion.

Decoding Features
$P(\mathbf{f}\|\mathbf{e})$
$P(\mathbf{e}\|\mathbf{f})$
$LexW(\mathbf{f}\|\mathbf{e})$
$LexW(\mathbf{e}\|\mathbf{f})$
Phrase Penalty
Lexical Backoff
Word Penalty
Distortion
$P(\mathbf{E})$ – 6-gram language model
Rescoring Features
$P_{rescore}(\mathbf{E})$ – 7-gram LM
$P_{class}(\mathbf{E})$ – 7-gram class-based LM
$P_{Model1}(\mathbf{F}\|\mathbf{E})$ – IBM model 1 translation probabilities

Table 2: *Independent models used in log-linear combination*

The `moses` decoder [14] was used for our baseline system. This system serves as the starting point for our all experiments submitted during this year's evaluation. As described in the following sections, we implemented several techniques for generating improved phrase tables and language models, and experimented with using these techniques both individually and in combination.

2.3. Cross-Entropy Filtering

Based on the success of cross-entropy training data filtering [15] in last year's evaluation [16], we have continued experimenting with the technique across different language pairs. We used a 3-gram language model based filter, and experimented with LM cross-entropy filtering on each of our systems across different language pairs.

Corpus	Before Filtering	After Filtering
TED	141,387	141,387
FrEn 10^9	24,116,560	824,698
UN	12,886,831	220,066
Europarl	2,007,723	76,554
News Commentary	137,097	1,735
TOTAL	39,289,598	1,264,441

Table 3: *Cross-entropy data sizes, using the minimum-perplexity method*

We train a language model on a random subset of the out-of-domain corpus, of the same size as the TED training data. We then sort all sentences in the corpus based on the difference between their cross-entropy given the out-of-domain model and their cross-entropy given the TED language model. The filtered data is taken to be the highest scoring N sentences. We chose the size N in two different ways. First, we simply chose N to be some specific fraction of the data (for example, 5%, 10%, 15%, and 20%). Alternatively, we used an automated approach [17] that uses an information-theoretic estimate of the data size. We train new language models on the best $1/64$, $1/32$, $1/16$, $1/8$, $1/4$, and $1/2$ of the corpus. We selected the filter size that produced the language model with the minimum perplexity on the `dev2010` dataset. To filter the parallel data, we combined the perplexity thresholds that produced the best source and target language models for the `dev2010` dataset.

In general, we aggregated together all out-of-domain parallel data, and performed filtering on the resulting set of sentences; however, we also ran an experiment where automatic filtering was performed independently on each out-of-domain corpus.

In English-to-French, when running automatic filtering on each corpus, this resulted in the selection of 3.2 percent of the overall data for translation model, as shown in Table 3. We also tried manual filtering settings. In all cases, the translation model was fully generated from all of the filtered data.

Our manual filtering results were tested on `tst2010`. In French, we tried 5%, 10%, and 15%. Starting from the system Contrast4 (See Sec 4), changing only the percentage of cross-entropy filtered data, we obtained mean BLEU scores of 31.78, 32.21, and 31.73, respectively. In Russian, we tried 10%, 20%, 30%, and obtained 16.74, 16.64, and 16.62, respectively, compared to a baseline score of 17.13. In Chinese, the same percentages yielded scores of 7.61, 7.37, and 6.57, which were all significantly lower than the baseline average of 10.93. In Arabic, we obtained 24.23, 23.42, and 23.79, with a baseline of 23.90. We stopped increasing the filter size when either performance significantly deteriorated, or our job scheduler terminated Moses (typically when it used more than 200 GB of resident memory).

A list of the corpora can be found in Table 4. In the Chinese and Arabic to English test sets, we used data from the Multi-UN corpora that we sentence aligned using Champollion [18] to obtain the results discussed above. Despite the reasonable sentence pairs produced, we found no significant improvement in the scores.

2.4. Improvements to Optimization

We introduce a new optimization technique, "Derivative-free robust error minimization", or DREM. It is distinguished from MERT by its (a) coordinate system, (b) objective function, and (c) other procedural features.

Corpus	Lang.	Num Sent
Europarl-v7	en-fr	1,495,313
UN	ar-en	4,743,378
UN	ru-en	8,344,467
UN	zh-en	5,948,155
UN	en-fr	9,018,500
10^9	en-fr	15,515,787
News Comm.	en-fr	107,756
Common Crawl	en-fr	2,563,465
Wiki Headlines	ru-en	512,000

Table 4: A list of the parallel out-of-domain data used in Cross Entropy filtering. Number of sentences is after filtering out sentences of length > 40.

Optimizer	tst2010	tst2011
DREM	32.82	39.35
MERT	32.41	-
PRO	32.79	39.37
Rampion	32.88	39.10

Table 5: Performance of different optimization methods in English-to-French, for the submission system configuration (system details described in Section 4)

With regard to the coordinate system, the weights are tuned on a variance-normalized multi-dimensional unit sphere, rather than in the standard Euclidean space. This incorporates the scale-invariance of the weights and reduces the dimension of the search space by one. Second, it randomizes the coordinate system at every step, which allows it to search in multiple random directions without multiplying the time and effort required.

There are two main novel features of the objective function minimized by DREM. First, the estimated decoder score at a new point considers how far away the new point is from the decode points. A translation that was produced at the closest decode will get full trust, and a translation produced at a more distant decode will be penalized. Second, DREM is not an exhaustive search of the error along a line. Instead, the error function is sampled around the current point and modeled by a (quadratic or linear) function via least-squares regression. This model is minimized around the point, subject to not moving too far away (a "trust-region" constraint). This both reduces metric computations and prevents a sharp valley or spike in the objective function from dominating the behavior, making the result more robust.

Finally, there are two main procedural features of DREM. First, the search for optimal weights is restarted at a few of the most promising of the past decode points, preventing a misstep at an early iteration from having a lingering effect. Second, we have control of the error function minimized. We can manipulate the n-best list into the desired format and use our choice of metrics to define the error function. For this competition, we transformed the n-best list into human-readable text and chose the error function $1 - \frac{1}{2}(\text{Expected BLEU score} + \text{Expected Meteor score})$.

We compare our results from using DREM on our best systems against MERT, and two other optimization techniques: (a) PRO (Pairwise Ranking Optimization) due to Hopkins and May [19], and (b) Rampion, a technique based on Structured Ramp Loss due to Gimpel and Smith [20]. The results per language can be seen in Tables 5,6,7,8.

Optimizer	tst2010	tst2011	tst2012	tst2013
DREM	19.39	21.46	19.28	21.57
MERT	19.24	21.24	19.30	21.70
PRO	19.67	21.32	19.61	21.71
Rampion	18.88	20.57	18.55	20.44

Table 6: Performance of different optimization methods in Russian-to-English.

Optimizer	tst2010	tst2011	tst2012	tst2013
DREM	11.32	15.74	13.91	14.60
MERT	11.13	14.12	12.28	13.21
PRO	11.87	15.34	13.45	14.52
Rampion	11.10	14.19	12.32	13.22

Table 7: Performance of different optimization methods in Chinese-to-English.

2.5. Language Modeling

For decoding, a number of different language models were used in various experiments. In general, the procedure was to train a single language model for each domain and subdomain. For example, we would obtain one language model for each news source of the English/French Gigaword corpora. We then either (a) interpolated several language models together, using the MITLM toolkit, or (b) let each language model have its own λ_i to be optimized by MERT/DREM/PRO/Rampion, or (c) some combination thereof. Specific submission details can be found Section 4. A list of the monolingual data used can be found in Table 9.

We rescored our n-best lists using both class language models (order-7) and recurrent neural network language models (RNNLM) [21]. The former were trained on the target side of the cross entropy filtered data, while the latter were trained on the monolingual TED data (train.fr/train.en). The recurrent neural network contained 160 hidden units, 300 classes and backpropagation through time of 4. Additionally, some of the Chinese-English systems used a second RNN that contained 10 hidden units and 100 classes. RNN was responsible for substantial gains in most cases. For a summary of its effects, see Table 10.

2.6. Lexical approximation

Morphologically rich languages pose a challenge for machine translation systems due to the high number of alternate forms each word may take. Particularly when the size of the training data is small, this creates a sparsity problem for word alignments and results in a higher out-of-vocabulary (OOV) rate. Without specific processing, unknown words are either output without being translated, or are omitted, both of which hurt translation quality. To translate these words, we utilized lexical approximation, which generates alignments for unknown words by approximating those of the closest known word in our GIZA++ word alignments [25].

For each OOV word, we find a series of most-likely candi-

Optimizer	tst2010	tst2011	tst2012	tst2013
DREM	25.03	25.68	27.65	26.79
MERT	24.71	25.27	27.48	26.50
PRO	24.88	25.52	27.39	26.97
Rampion	24.05	24.83	26.53	25.83

Table 8: Performance of different optimization methods in Arabic-to-English.

Corpus	English	French
Europarl-v7	55,730,697	61,888,789
News Commentary	3,404,297	4,928,120
NewsCrawl '07-'11	2,309,306,270	616,057,716
FrGigaword v2	N/A	827,241,410
EnGigaword v5	4,195,862,612	N/A
UN	361,878,283	421,687,471
TED	2,719,842	2,800,512
10^9	668,269,385	810,599,307

Table 9: Summary of monolingual training data used.

Test set	RNN?	French	Russian	Chinese	Arabic
tst2010	N	32.34	19.34	11.24	25.46
	Y	32.82	19.39	11.32	25.49
tst2011	N	38.45	21.14	15.72	26.37
	Y	39.35	21.46	15.74	26.23
tst2012	N	N/A	19.33	13.92	28.41
	Y	N/A	19.28	13.91	28.47
tst2013	N	N/A	21.41	14.65	28.09
	Y	N/A	21.57	14.60	28.21

Table 10: Performance of various systems with and without Recurrent Neural Network language model rescoring. Scores are average BLEU over 10 iterations.

dates from word alignments utilizing character-based Levenshtein distance. We experimented with "approximating" only the in-vocabulary word with the minimum edit distance (1), and the set under a particular threshold (2). In the latter case, we weighted the probabilities of each alignment by the edit distance between the OOV word and its in-vocabulary approximation.

Table 11 shows the reduction in OOV words and the resulting performance improvement by using the above techniques. Due to its higher OOV rate, RU-EN translation benefited more than AR-EN.

	Russian		Arabic	
Processing	OOV rate	BLEU	OOV rate	BLEU
None	2.8%	21.85	2.2%	24.08
LA (1)	0.2%	21.86	0.1%	24.08
LA (2)	0.0%	21.98	0.1%	24.11

Table 11: OOV Rate and Mean BLEU scores for LA on tst2013.

As seen in table 11, LA with a set of values under a threshold performs better than a single replacement, though both provide minor improvement over not processing OOV words. These results are likely due to the fact that while edit distance finds close word forms, there is no guarantee that similar word forms have similar alignments. Further, an OOV word may have no truly similar words in our vocabulary, making its approximation unrelated. In this light, thresholding a set of values provides more possibilities from which a likely alignment to arise.

2.7. Development set selection

In past evaluations we have always used the development data given to tune the parameters of our system; however, there is no reason to suspect that tuning performance is independent of the data used, nor that the given TED talks will produce optimal weights for decoding. We try using alternative data for tuning, extracted directly from the TED training data. The sentences not chosen are used for the normal training procedure.

System	tst2010	tst2011
En-Fr	29.11	35.42
En-Fr + Dev	29.23	35.74
En-Fr C4	32.01	38.75
En-Fr C4 + Dev	32.01	39.22
Ru-En	18.71	21.17
Ru-En + Dev	18.61	20.44
Zh-En	11.27	14.22
Zh-En + Dev	10.31	14.49

Table 12: Results with and without dev set selection, using tst2010 as a target. Scores are average BLEU over 10 iterations, case+punc. C4 refers to the submitted system "Contrastive 4."

Ideally, a development set should resemble the data one expects to decode. Given a language model describing the expected test data, the development set should be drawn from the same distribution. dev2010 and all evaluation data sets are TED talks, so this is loosely the case already, but we investigate further refinement of the development set. We built a selection algorithm that, given a test set as input, extracts the most similar subset of the TED training data.

Since language models are based off of n-gram counts, our algorithm samples from among the training data to match overall n-gram count frequency. Our algorithm samples to minimize an objective function that loosely resembles the KL-divergence between two language models. (In future work, we will use discounting and explicitly minimize KL divergence.) Let S and T refer to the selection set and input test set, and let $C_S(j)$ indicates the count of n-gram j in the selection set, $C_T(j)$ the analogous count in the test data. The objective function $F(S,T)$ we used is:

$$F(S,T) = \sum_j a\left(\log\frac{C_S(j)}{C_S} - \log\frac{C_T(j)}{C_T}\right)$$
$$a(x) = \text{if } x > 0 \text{ then } x \text{ else } -\tfrac{1}{3}x$$

This pseudo absolute-value $a(x)$ is used to penalize spurious n-grams less than missing n-grams. We tracked n-grams up to order 3, and missing counts in the above formula were given the value 0.1. Table 12 gives the performance of this algorithm on several experiments.

3. MT Language Specific Algorithms

3.1. Arabic-to-English Morphological Processing

In our Arabic-to-English MT systems for prior year evaluations [22, 23, 24, 25, 26], we normalized various forms of alef and hamza and removed the tatweel character and some diacritics before applying a light Arabic morphological analysis procedure that we called AP5. Last year, [3] we modified the AP5 procedure to more closely conform to the Arabic Treebank (ATB) segmentation format used in the MADA Arabic morphological analysis, diacritization, and lemmatization system, [27]. This year, we compared the AP5 system to MADA directly, seen in Table 15.

All systems with the rule-based MADA+TOKAN processing outperformed the same system on all test sets with AP5. The degree depended both on the test set and on the optimizer, as seen in Table 15. The most signifant gains were seen using MERT, with a 1.34 BLEU improvement over AP5 on tst2013. While both analyses regularize affixes and perform stemming, MADA more pervasively normalizes character variation and segments more heavily than AP5, reducing the OOV rate from 7.0% with AP5 to 2.2% with MADA.

Segmenter	BLEU
charSeg	10.37
cmuSeg	9.78
stanSegCTB	10.72
stanSegPKU	10.58
charSeg+cmuSeg	10.71
charSeg+stanSegCTB	10.83
charSeg+stanSegPKU	10.66

Table 13: Comparison of baseline MT systems for Chinese-English based on vaious word segmenters. The BLEU score is an average over 10 experiments for `tst2010`.

3.2. Chinese-to-English Character and Word Segmentation

One challenge of building a machine translation system for Chinese is the absence of spaces between words. We trained systems based on a few different word segmenters for the machine translation task and selected the top performer based on average BLEU score to be our baseline system for this evaluation. The results of our comparison are in Table 13.

The Stanford Chinese Word Segmenter [28] was evaluated using both the Chinese Penn Treebank (CTB) and the Peking University (PKU) segmentation standards. In addition, the CMU LDC Word Segmenter [30] and simply segmenting each individual character were evaluated. GIZA++ was trained using sentences from each segmentation result. Next, the alignment file for each segmenter was further character segmented and combined with the GIZA++ alignments from the character segmenter before being used to create the phrase table.

The Stanford CTB segmenter out-performed the other individual segmenters, and we saw additional gains from combining GIZA++ alignments for this segmenter with the character segmented GIZA++ alignments. As a result, we chose to use the char+stanCTB segmenter for this evaluation.

3.3. Russian-to-English Morphological Segmentation

To compensate for the morphological complexity in Russian, we experimented with segmentation. We utilized Morfessor Cat-MAP both to process all the data as well as only for word alignments (WA), [29]. Table 14 shows the mean BLEU scores for individual Russian-to-English MT systems trained on the 2013 training data and tested on the 2010 test set. Morfessor categorizes proposed segments as prefixes, stems, or suffixes. We both kept all generated segments, as well as only stems.

	RUSSIAN	
Processing	OOV rate	BLEU
None	5.0%	17.26
Stems for WA only	3.3%	16.44
Morfessor Stems	4.5%	16.15
Morfessor All Segs	2.4%	16.54

Table 14: Russian-Specific Experiments, OOV rate and BLEU scores for `tst2010`.

Though processing the data with Morfessor decreased the OOV rate by up to 51.6%, BLEU score decreased. Though word alignments were improved, it was more difficult to organize a greater number of target words into meaningful sentences. Before segmentation, source sentences had on average 14.2 tokens per sentence against 17.12 for English, the relation we would expect given the morphological complexity of Russian. With the best segmentation result (see Table 14) we have 19.3 tokens per Russian sentence. An explanation for poorer performance, then, is that instead of bringing sentence lengths closer together and making fertility closer to 1:1, segmentation widened the gap between the two languages.

4. MT Submission Summary

The different experiments we ran in Sections 2 and 3 of this paper played different roles in the submission systems of different languages. In this section we describe the systems that were submitted, and their respective scores. In the tables that follow, the following abbreviations are used:

- **lexDist**: Refers to the moses lexicalized reordering model `wbe-msd-bidirectional-allff`
- **dunk**: Drop unknown words
- **(corpus 1)**$\cdots$**(corpus n) LM**: Linear interpolation of several LMs
- **RNNx**: RNN order x
- **FiltLM**: Data for language model filtered via Cross Entropy with TED LM (not interpolated)
- **LA**: Lex approx

System combinations were trained using the `tst2010`. We therefore omit scores for `tst2010` on those systems.

It is also worth noting that systems trained at MITLL used manual cross entropy filter sizes, while those at AFRL used minimum perplexity threshold filter sizes. This is mentioned in the discussion sections.

4.1. English-to-French

For French, our best system for `tst2013` (which was submitted as contrastive) used a single order 5 language model from the MITLM toolkit, consisting of the following LMs linearly interpolated (using the MITLM toolkit) on `dev2010`: TED, Europarl-v7, News-Commentary-v7, News-Crawl2007, News-Crawl2008, News-Crawl2009, News-Crawl2010, News-Crawl2011, and the 10^9 corpus. We found inclusion of LDC French Gigaword v2 did not improve the performance of this language model. The phrase table was filtered at 10% extra data using cross entropy, without using Common Crawl. Our other French systems used a combination of a 6th-order TED language model, and a linearly interpolated language model over LDC Gigaword v2, Europarl, and News Commentary data set. Results are in Table 15. The phrase tables were obtained using cross-entropy filtering with minimum perplexity thresholds on each of the data sets, and including Common Crawl.

4.2. Chinese-to-English

Table 15 describes each of the systems we submitted for the Chinese-English portion of the machine translation task. Our primary system is a combination of four different systems.

The best-scoring single system on the `tst2010` data set was the PRO-optimized system, so we decided to set the system combination weights to favor the PRO system over the others. However, the DREM system scored the best for the other data sets. Our contrastive2 submission had a significantly higher weight for the PRO system compared to the weight for our primary submission. Perhaps we would have seen even higher scores for the `tst2013` data set if we had set the weights higher for the DREM system. When performing system combination, the primary and contrastive2 systems used different prior weights during training.

4.3. Russian-to-English

Table 15 describes each of the systems we submitted for the Russian-English portion of the machine translation task. Our primary system is a combination of three different systems. Our best system on tst2013, improperly tokenized when submitted as Contrast3†, used a single 4th-order language model from the MITLM toolkit, consisting of the following LMs linearly interpolated (using the MITLM toolkit) on dev2010: TED, MultiUN, Wikipedia headlines, and LDC English Gigaword v5. The phrase table kept 20% of extra data using cross entropy filtering, and used Wikipedia Headlines + United Nations data. Our other Russian systems used a combination of a 6th-order TED language model, and a linearly interpolated language model over the LDC Gigaword, MultiUN, and News-Crawl2007, News-Crawl2008, News-Crawl2009, News-Crawl2010, News-Crawl2011 corpora. For the cross entropy filtering, we used News-Commentary-v7 and the News-Crawl corpora with minimal perplexity thresholds.

4.4. Arabic-to-English

Before the deadline, we were only able to submit results for AP5 with various optimization. However, we include in the table results for MADA, which significantly outperforms the submitted systems.

5. Automatic Speech Recognition

Acoustic training data for our ASR systems were harvested from 838 TED Talks. We applied the same alignment and closed caption filtering process as IWLST 2011 [26], except that each utterance was padded by a maximum of 0.25 seconds (instead of 2.0 seconds) and the filtering threshold was set to 30% WER (Word Error Rate). This yielded 166 hours of audio.

A GMM-HMM system was trained using Perceptual Linear Prediction (PLP) features. This system was developed using the same training procedure as our IWSLT 2011 system, except that this year we applied mean and variance feature normalization on a per speaker basis. The updated data partition and feature normalization yielded a 1.0% WER reduction on dev2010, tst2010, and dev2012.

A secondary GMM-HMM (GMM-HMM-2) system was trained in a similar fashion as the prior, but using the CMU Pronouncing Dictionary [31]. Missing dictionary entries from the training data were generated by training a grapheme-to-phoneme model using Sequitur G2P, an open-source grapheme-to-phoneme converter [32]. This system did not quite reach the performance of the other GMM-HMM system, however, use of the stress markings included in the CMU Pronouncing Dictionary are being furthered explored to evaluate their impact on performance, and initial tests show a decrease in WER on dev2010 of approximately 0.3% as compared to ignoring the stress markings.

A hybrid Deep Neural Network (DNN)-HMM speech recognition system was developed using Theano [33] and a version of HTK that we modified according to the method of [34]. The DNN included 5 hidden layers, each of which had 1000 neurons with logistic activation functions. A context window of 9 frames was used at the input, and the output included 6000 units corresponding to the shared states of our GMM-HMM system. The feature set consisted of 13 PLPs with delta and acceleration coeffcients, and all features were normalized to zero mean and unit variance on a per speaker basis. Training was performed using layer growing back propagation [35] with a minibatch size of 512, and an initial learning rate of 0.008 that was halved after each epoch once the improvement in accuracy on the cross validation partition fell below 0.5%. A second DNN was trained on PLP features that were transformed using Constrained Maximum Likelihood Linear Regression (CMLLR). This system applied a single transform per speaker.

LM data selection was implemented using the same procedure as our IWSLT 2012 system [3]. Interpolated trigram and 4-gram LMs were estimated on TED, 1/8 of Gigaword, and 1/4 of News 2007–2012 using the SRILM Toolkit.[1] Compared to a trigram LM trained on all of the available data, applying data selection reduced the WER of our GMM-HMM system by 0.4% on dev2010, tst2010, and dev2012. Recurrent Neural Network Maximum Entropy (RNNME) LMs were trained on 1/16 of Gigaword and 1/8 of News 2007–2012 using the RNNLM Toolkit [21]. Each network included 160 hidden units, 300 classes in the output layer, 4-gram features for the direct connections, and a hash size of 10^9. The LM vocabulary included 95000 words.

A neural network based Speech Activity Detector (SAD) was developed using Theano. The SAD was trained on 22 hours of TED data and 5 hours of public domain music downloaded from Wikimedia Commons,[2] the United States Air Force Band,[3] and the Open Goldberg Variations project.[4] The network included a context window of 21 frames on the input, 1 hidden layer of 500 neurons with logistic activation functions, and 3 output units corresponding to speech, silence/noise, and music. The feature set consisted of 13 PLPs with delta and acceleration coeffcients, and all features were globally normalized to zero mean and unit variance. Training was perfomed using the same procedure as the DNNs.

Automatic segmentation of the test data was performed by evaluating the SAD, applying a dynamic programming algorithm to choose the best sequence of states, and padding the speech end points by 0.15 seconds. The speech segments from each talk were clustered using the MIT-LL GMM-based speaker recognition software package. Compared to the manual segmentation provided in the reference files, automatically segmenting the test data increased the WER of our GMM-HMM system by 0.7% on dev2010, tst2010, and dev2010.

Initial transcripts of the test data were produced using the hybrid DNN-HMM system. Next, CMLLR transforms were estimated for the GMM-HMM system and the second hybrid DNN-HMM system. Recognition lattices were produced for each system and then rescored with the interpolated 4-gram LM. The final transcripts were produced by rescoring n-best lists with the RNNME LMs.

System combination was performed as in IWSLT 2012 [3] using a Confusion Network Combination system (CNC). Confusion networks for combined systems are generated from rescored n-best lists of size 1000. The confusion networks are aligned with each other, and this alignment is used to merge the individual system's confusion networks into one. Each system is weighted (weights generated from a Powell-like grid search) and acoustic model and language model scores of each are combined. Due to time constraints, tests for our system combination were performed on dev2010 using the GMM-HMM and DNN-HMM, but results were not submitted. Table 16 shows WERs for the individual systems and WER for the combined system using this methodology.

Table 17 shows results on dev2010 for the DNN-HMM system, the GMM-HMM system, and the GMM-HMM-2 system before and after RNNLM rescoring. Table 18 shows the progress of our current systems against our best submission from IWSLT 2012 [3] on tst2011 and tst2012. Results for tst2013 on our current submissions are also shown.

[1] Available at: http://www.speech.sri.com/projects/srilm
[2] Available at: http://commons.wikimedia.org
[3] Available at: http://www.usafband.af.mil
[4] Available at: http://www.opengoldbergvariations.org

System	Description	tst2010	tst2011	tst2012	tst2013
English-to-French					
primary	DREM + UNTen9EuroNCommCC FiltLM + Giga LM + RNN3 + dunk	32.82	39.35	39.76	37.05
contrast1	Ramp + UNTen9EuroNCommCC FiltLM + Giga LM + RNN3 + dunk	32.88	39.10	39.94	37.12
contrast2	Primary + Contrast4	N/A	38.97	39.70	37.32
contrast3	PRO + UNTen9EuroNCommCC FiltLM + Giga LM + RNN3 + dunk	32.79	39.37	39.70	37.41
contrast4	MERT + tedUNTen9NewsCrawlEuro LM + dunk	32.21	38.90	39.83	37.58
contrast5	Primary + Contrast3	N/A	39.27	39.97	37.21
Chinese-to-English					
primary	contrast5 + contrast3 + contrast6 + contrast4	11.46	15.92	14.05	14.85
contrast1	contrast5 + contrast3 + contrast4	11.40	15.90	13.59	14.77
contrast2	contrast5 + contrast3 + contrast6 + contrast4	11.53	16.00	14.00	14.77
contrast3	PRO + tedGiga LM + RNN3 + lexDist + dropunk	12.03	15.14	13.50	14.36
contrast4	MERT + tedUNGiga LM + RNN5	11.72	14.64	12.35	13.25
contrast5	DREM + tedLM + Giga LM + RNN3 + lexDist + dunk	11.47	15.85	13.93	14.61
contrast6	MERT + tedUNGigaEuroNComm LM + RNN5 + dunk	11.20	14.87	12.63	13.50
Russian-to-English					
primary	contrast1 + contrast2 + contrast3	N/A	21.49	19.61	21.65
contrast1	PRO + tedNewsCrawlCC FiltLM + Giga LM + RNN3 + LA2 + dunk	19.67	21.32	19.61	21.71
contrast2	MERT + tedNewsCrawlCC FiltLM + Giga LM + RNN3 + LA2 + dunk	19.24	21.24	19.30	21.70
contrast3†	MERT + tedUNWiki LM + LA2 + dunk	19.42	21.68	19.58	22.13
contrast4	DREM + tedNewsCrawlCC FiltLM + Giga LM + RNN3 + LA2 + dunk	19.39	21.46	19.28	21.57
Arabic-to-English					
primary	DREM + lexDist + ted LM + giga LM + RNN3 + AP5 + dunk	25.03	25.66	27.66	26.64
contrast1	PRO + lexDist + ted LM + giga LM + RNN3 + AP5 + dunk	24.88	25.81	27.52	27.27
contrast2	MERT + lexDist + ted LM + giga LM + RNN3 + AP5 + dunk	24.71	24.95	27.27	26.22
contrast3	MERT + lexDist +tedUNGigaEuro LM + AP5 + dropunk	24.36	24.96	26.95	25.77
MADA	DREM + lexDist + ted LM + giga LM + RNN3 + MADA + dunk	25.49	26.23	28.47	28.21
MADA1	PRO + lexDist + ted LM + giga LM + RNN3 + MADA + dunk	25.21	26.41	27.92	27.70
MADA2	MERT + lexDist + ted LM + giga LM + RNN3 + MADA + dunk	25.14	26.01	27.97	27.56

Table 15: All Submission Systems. †For this system, fixed tokenization issue after submission.

dev2010		
DNN-HMM	GMM-HMM	Combined
13.9	14.5	13.7

Table 16: WER for individual DNN-HMM and GMM-HMM systems and their system combination on dev2010 (automatic segmentations, without RNNLM rescoring).

dev2010		
	without RNNLM	with RNNLM
DNN-HMM	14.1	12.9
GMM-HMM	13.8	12.8
GMM-HMM-2	17.2	15.6

Table 17: WER without and with RNNLM rescoring on dev2010 (manual segmentations).

	tst2011	tst2012	tst2013
IWSLT 2013			
DNN-HMM	10.6	11.3	15.9
GMM-HMM	9.7	11.0	16.7
GMM-HMM-2	12.5	13.9	23.0
IWSLT 2012			
GMM-HMM	12.6	14.3	N/A

Table 18: WER of IWSLT 2013 submissions on tst2011, tst2012 versus our best 2012 system. tst2013 is also shown.

6. Acknowledgments

We would like to thank members of the Wright-Patterson AFB, and of the Human Language Technology group at MIT Lincoln Lab for their support and state of the art computer systems.

7. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, S. Stüker "Overview of the IWSLT 2012 Evaluation Campaign," In *Proc. of IWSLT*, Hong Kong, HK, 2012.

[2] M. Cettolo, C. Girardi, and M. Federico "WIT3: Web Inventory of Transcribed and Translated Talks," In *Proc. of EAMT*, pp. 261-268, Trento, Italy, 2012.

[3] J. Drexler, W. Shen, T. Gleason, T. Anderson, R. Slyh, B. Ore, and E. Hansen, "The MIT-LL/AFRL IWSLT-2012 MT System," in *Proceedings of IWSLT 2012,* (Hong Kong, HK), 2012.

[4] Shen, W., Delaney, B., and Anderson, T. "The MIT-LL/AFRL IWSLT-2006 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Kyoto, Japan, 2006.

[5] G. Foster, R. Kuhn, and H. Johnson, "Phrasetable smoothing for statistical machine translation," in *Proceedings of EMNLP 2006,* (Sydney, Australia), July 2006.

[6] Chen, B. et al, "The ITC-irst SMT System for IWSLT-2005,"

In Proc. Of the International Workshop on Spoken Language Translation, Pittsburgh, PA, 2005.

[7] Brown, P., Della Pietra, V., Della Pietra, S. and Mercer, R. "The Mathematics of Statistical Machine Translation: Parameter Estimation," Computational Linguistics 19(2):263–311, 1993.

[8] Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, I.D., Och, F.J., Purdy, D., Smith, N.A., Yarowsky, D., "Statistical machine translation: Final report," In Proceedings of the Summer Workshop on Language Engineering at JHU, Baltimore, MD 1999.

[9] Och, F. J. "An Efficient Method for Determining Bilingual Word Classes," Ninth Conf. of the Europ. Chapter of the Association for Computational Linguistics; EACL'99, pp. 71-76. Bergen, Norway, June 1999.

[10] Bo-June (Paul) Hsu and James Glass, "Iterative Language Model Estimation: Efficient Data Structure and Algorithms," In Proc. Interspeech, 2008.

[11] Melamed, D., "Models of Translational Equivalence among Words," In Computational Linguistics, vol. 26, no. 2, pp. 221-249, 2000.

[12] Liang, P., Scar, B., and Klein, D., "Alignment by Agreement," Proceedings of Human Language Technology and North American Association for Computational Linguistics (HLT/NAACL), 2006.

[13] Och, F. J., "Minimum Error Rate Training for Statistical Machine Translation," In ACL 2003: Proc. of the Association for Computational Linguistics, Japan, Sapporo, 2003.

[14] Koehn, P., et al, "Moses: Open Source Toolkit for Statistical Machine Translation," Annual Meeting of the Association for Computational Linguistics (ACL), Prague, Czech Republic, June 2007.

[15] S. Mansour *et al*., "Combining Translation and Language Model Scoring for Domain-Specific Data Filtering," in *Proc. International Workshop on Spoken Language Translation*, San Francisco, USA, 2011.

[16] E. Hasler *et al*., "The UEDIN Systems for the IWSLT 2012 Evaluation," in *Proceedings of IWSLT 2012,* (Hong Kong, HK), 2012.

[17] R. C. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *Proceedings of the ACL 2010 Conference Short Papers,* (Uppsala, Sweden), 2010.

[18] X. Ma, "Champollion: A Robust Parallel Text Sentence Aligner," in *Proceedings of LREC 2006,* (Genova, Italy), 2006.

[19] M. Hopkins and J. May, "Tuning as ranking," in *Proc. of EMNLP 2011* (Edinburgh, Scotland, UK), July 2011.

[20] K. Gimpel and N. A. Smith, "Structured Ramp Loss Minimization for Machine Translation," in *Proceedings of NAACL 2012,* (Montreal, Canada) June 2012.

[21] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černocký, "Strategies for Training Large Scale Neural Network Language Models," in *Proc. Automatic Speech Recognition and Understanding Workshop*, Hawaii, USA, 2011.

[22] Shen, W., Delaney, B., Anderson, T., and Slyh, R. "The MIT-LL/AFRL IWSLT-2007 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Trento, Italy, 2007.

[23] Shen, W., Delaney, B., Anderson, T., and Slyh, R. "The MIT-LL/AFRL IWSLT-2008 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Honolulu, HI, 2008.

[24] Shen, W., Delaney, B., Aminzadeh, A.R., Anderson, T., and Slyh, R. "The MIT-LL/AFRL IWSLT-2009 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Tokyo, Japan, 2009.

[25] Shen, Anderson, T., Slyh, R., and Aminzadeh, A.R., "The MIT-LL/AFRL IWSLT-2010 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Paris, France, 2010.

[26] A. R. Aminzadeh, T. Anderson, R. Slyh, B. Ore, E. Hansen, W. Shen, J. Drexler, and T. Gleason, "The MIT-LL/AFRL IWSLT-2011 MT system," in *Proceedings of IWSLT 2011,* (San Francisco CA), December 2011.

[27] R. Roth, O. Rambow, N. Habash, M. Diab, and C. Rudin, "Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking," in *Proceedings of ACL-08: HLT, Short Papers,* (Columbus OH), June 2008.

[28] P. C. Chang, M. Galley, and C. D. Manning, "Optimizing Chinese Word Segmentation for Machine Translation Performance," in *Proceedings of the Third Workshop on Statistical Machine Translation,* (Columbus, OH), June 2008.

[29] M. Creutz, K. Lagus. "Unsupervised models for morpheme segmentation and morphology learning," in *ACM Transactions on Speech and Language Processing*, 4(1):1-34, 2007.

[30] Z. Wu. "LDC Chinese Segmenter," *http://www.ldc.upenn.edu/Projects/Chinese/segmenter/mansegment.perl*, 1999.

[31] R. Weide., "The CMU pronouncing dictionary," *URL: http://www. speech. cs. cmu. edu/cgibin/cmudict*, 1998.

[32] M. Bisani., "Sequitur G2P: A trainable Grapheme-to-Phoneme converter," *http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html*, 2011.

[33] J. Bergstra *et al*., "Theano: A CPU and GPU Math Expression Compiler," in *Proc. Python Scientific Computing Conference (SciPy)*, Austin, TX, 2010.

[34] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, January, 2012.

[35] F. Seide, G. Li, and D. Yu, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks," in *Proc. Interspeech*, Florence, Italy, 2011.

The Speech Recognition and Machine Translation System of IOIT for IWSLT 2013

Ngoc-Quan Pham, Hai-Son Le
Tat-Thang Vu, Chi-Mai Luong

Institute of Information and Technology (IOIT),
Vietnamese Academy of Science and Technology (VAST)
(quanpn,lehaison,vtthang,lcmai)@ioit.ac.vn

Abstract

This paper describes the Automatic Speech Recognition (ASR) and Machine Translation (MT) systems developed by IOIT for the evaluation campaign of IWSLT2013. For the ASR task, using Kaldi toolkit, we developed the system based on weighted finite state transducer. The system is constructed by applying several techniques, notably, subspace Gaussian mixture models, speaker adaptation, discriminative training, system combination and SOUL, a neural network language model. The techniques used for automatic segmentation are also clarified. Besides, we compared different types of SOUL models in order to study the impact of words of previous sentences in predicting words in language modeling. For the MT task, the baseline system was built based on the open source toolkit N-code, then being augmented by using SOUL on top, i.e., in N-best rescoring phase.

1. Introduction

This paper describes the two systems developed by IOIT, serving the two tasks in the IWSLT 2013 evaluation campaign, namely Automatically Speech Recognition (ASR) and Machine Translation (MT).

The English ASR task focuses on translating TED talks which are a collection of public lectures on a variety of topics, ranging from Technology, Entertainment to Design. Apparently, the hindrances in the track are the spontaneous and natural way of speech, interruption of invalid noises such as music or applauses or dealing with topic adaptation. This year, since the evaluation data is no longer provided with manual sentence segmentation, dividing the long audio files into short utterances properly becomes a new challenging obstacle. For this task, we use Kaldi [1] to construct the system based on state-of-the-art techniques, notably, subspace Gaussian mixture models, speaker adaptation, discriminative training, system combination and SOUL [2], a neural network language model (NNLM). Finally, the system is a combination of two systems differing in acoustic model, augmented by rescoring the output N-best list with SOUL language models. Besides, we study the impact when SOUL language models take into account words of previous sentences in the context.

On the English to French MT task, since it is our first participation, our aim is to build a whole system from scratch using open source toolkits for normalization, tokenization, tagging, data filtering, system construction... which will be served as a baseline system for future research. The system is based on N-code[1], a bilingual n-gram approach for MT and the use of SOUL in N-best rescoring.

The organization of the paper is as follows: Section 2 is the description of our ASR system. While acoustic model training procedure is presented in Section 2.1, the automatic segmentation process is described in Section 2.2. The language modeling with three types of SOUL models are described in Section 2.3. Then, in Section 2.4, the decoding procedure will be presented in detail. Section 2.5 is devoted to ASR experimental results and our analyses . Section 3 is concentrated on the MT task. It consists of three parts: Section 3.1 for data preprocessing, Section 3.2 for the description of our system and Section 3.3 for the experimental evaluation.

2. Automatic Speech Recognition Task

2.1. Acoustic Modeling

2.1.1. Training corpus

We decided to collect TED lectures as training materials, in order to guarantee the homogeneity of training and development data in terms of speaking environment and speaking style. Approximately 220 hours of audio, distributed among 920 talks, were crawled with their subtitles, which were deliberately used for making transcripts. However, the provided subtitles do not contain the correct time stamps corresponding with each phrase as well as the exact pronunciation for the words spoken, which lead to the necessity for long-speech alignment.

Proved to be effective for long-speech alignment task, SailAlign [3, 4] is applied to extract text-aligned speech segments, which helps us to not only acquire the transcript with exact timing, but also to filter non-spoken sounds such as music or applauses. A part of these noises are kept for noise training while most of them are abolished. After that, the re-

[1] http://ncode.limsi.fr

mained audio used for training consists of around 175 hours of speech, distributed among nearly $175K$ utterances.

The lexicon was built based on the Carnegie Mellon University (CMU) Pronouncing Dictionary v0.7a, in which the phoneme set contains 39 phonemes and the word set contains 131,137 words. The vowels may also vary in lexical stress, ranging from no stress, primary stress to secondary stress.

2.1.2. Front-end

The front-end of the system is based on the conventional Mel-frequency cepstral coefficients (MFCC) features. The initial feature vectors, which contain 39 coefficients including 12 cepstral coefficients, 1 energy coefficient added with delta and double-delta features were extracted after windowing with the window size of 25 milliseconds and frame shift of 10 milliseconds. After that, Ceptral Mean and Variance Normalization (CMVN) was applied for normalization.

2.1.3. Training Procedure

The acoustic models were based on Hidden Markov Model (HMM), using Gaussian Mixture Models (GMM) for emission probabilities. In order to model context dependency, we used the tri-phone setup, with three states per phoneme and the topology was left-to-right. The model was trained with the expectation-maximization (EM) algorithm with a splitting procedure according to the maximum likelihood criterion. After splitting, the total number of gaussians, which are initiated at 2000, reached 200000. Furthermore, Maximum Likelihood Linear Regression (MLLR) technique was used to adapt the acoustic models with speaker information, for which we assumed that each TED talk in the training data is corresponding to one speaker.

Figure 1 reveals that we developed the systems in two directions after the baseline. On one hand, the acoustic model was strengthened throughout further training with subspace GMM, which was proved to significantly increase the system performance [5]. The SGMM model was then enhanced with discriminative training, producing the SGMM-MMI system. On the other hand, feature space discriminative training was implemented on top of the baseline system, to create the fMMI system. In order to display the progressive result, the error rates on dev2010 and tst2010 data are illustrated in Table 1. It is notable that the language model used in the experiments is the 3-gram LM described in Section 2.3. The SGMM was able to improve the performance of our system by 8% relatively, while discriminative training on top of the SGMM system showed its effectiveness by reducing the error rates by 13%. Feature space MMI training over the baseline system was efficient enough to reduce 18% of errors relatively. In brief, the SGMM+MMI training on top of the baseline system was slightly better than the counterpart trained with fMMI.

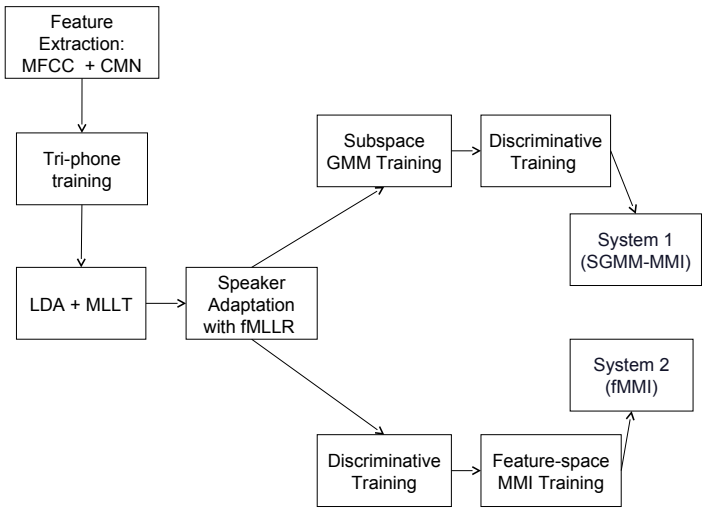


Figure 1: *Training Procedure diagram*

Table 1: *Progressive results shown by consecutively trained systems*

System	WER	
	dev2010	tst2010
MFCC+LDA+SAT (baseline)	26.6	26.4
baseline+SGMM	24.9	24.2
baseline+SGMM+MMI	21.8	21.1
baseline+fMMI	21.9	21.6

2.2. Auto-segmentation

Since the evaluation data in 2013 is no longer provided with timing information for segmentation, we utilize the LIUM Diarization toolkit [6] in order to divide the talk into small sentence-like segments,

Figure 2 provides a general description on the diarization process. First, 13 MFCC features are extracted from the long audio file. Subsequently, the long talk is segmented based on Viterbi Decoding, producing shorter segments which are at least 20 seconds long. After that, 8 one-state HMMs are used to remove music and jingle regions, leaving only speech segments. Detection of gender and bandwidth is then done using a GMM for each of the 4 combinations of gender (male / female) and bandwidth (narrow / wide band). Finally, GMM-based speaker clustering is carried out to map each speech segment to the corresponding speaker. Apparently, one TED talk can be given by only one or several speakers.

The disparity in word error rates is disclosed in Table 4, in Section 2.4. It is notable that the automatic speech detection caused approximately 2 percent loss of the spoken audio, resulted in inevitably decreasing the error rates, presented by deletions. Experiments conducted with tst2010 and dev2010 data illustrated that the WER increased 10% relatively, compared with the same data sets which are manually segmented.

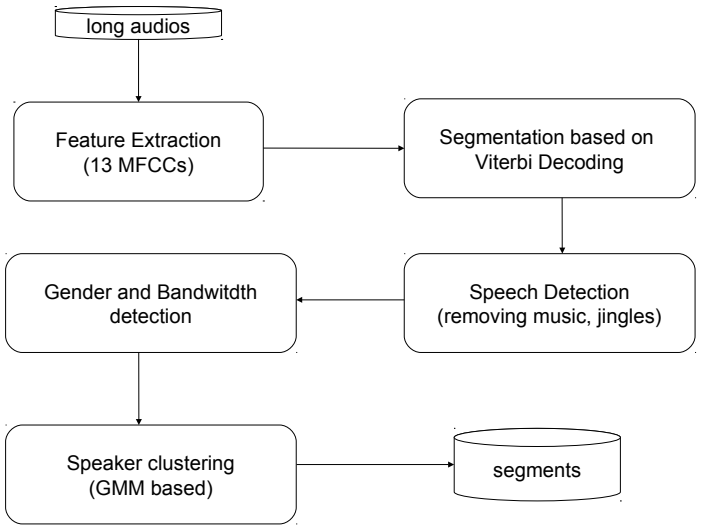


Figure 2: *Diarization Process.*

Due to the fact that the segmentation cannot be guaranteed to be precise at the beginning (end) of the sentence, the output segments are almost incomplete sentence, or incomplete phrases, which affects recognition results. The influence of language models on this problem will be analyzed later in Section 2.3.2.

2.3. Language Modeling

2.3.1. Overview

We used the in-domain data provided by organizer. In addition, we utilize $\frac{1}{8}$ of Giga corpus by filtering it according to the Moore-Lewis approach [7]. Both two datasets were normalized using the normalization toolkit from CMU[2]. The statistics of training data is summarized in Table 2. The vocabulary used to train language models is the same as in the lexicon. It consists of 131,137 words.

Table 2: *Training data for language modeling for English ASR Task*

Data	Number of sentences	Number of tokens
TED	156,460	2,708,816
$\frac{1}{8}$ Giga	2,565,687	56,488,064

The final model is the combination of two models trained on these datasets using SRILM toolkit with the modified interpolated Knesey-Ney smoothing technique [8].

Simultaneously, we trained SOUL language models on the same training data following exactly the procedure described in [9]. We use 300 as the dimension projection, $600; 300$ as the size of 2 hidden layers and $1000; 1000$ as the size of the shortlist and the number of classes for the out-of-shortlist words. For each type of SOUL models presented below, only one model is trained and used while decoding.

[2]http://www.festvox.org/nsw/

2.3.2. Auto-segmentation and sentence boundary problem

As auto-segmentation presented in Section 2.2 is based solely on acoustic features, each resulting segmentations does not correspond to a "normal" sentence but rather a phrase. For example, in dev2010, the audio for this sentence:

Now there are many of us who sort of forget that when I say...

is segmented into three parts corresponding to:

Now
there are many of us who sort of
forget that when I say...

If we train language models on data containing normal sentences, there will be a mismatch between test data and training data. The question is to what extent this mismatch affects the final performance. To partially answer this question, we proposed to use three types of SOUL models that differ in the way of treating sentence boundary. The detailed explanation of each model will be presented as follows:

Standard model Supposing that we use 4-gram language models and have a couple of sentences in a document:

Music can be the food of love
Let's do this

In the traditional way, the probability of the second sentence is:

$$p(\text{Let's}|\text{<s> <s> <s>}).p(\text{do}|\text{ <s> <s> Let's}).$$
$$p(\text{this}|\text{<s> Let's do}).p(\text{</s>}|\text{Let's do this}), \quad (1)$$

where <s>, </s> stand for the start (end) of the sentence. <s> is repeated at the beginning of the sentence to better represent the context in SOUL structure because the number of input tokens of SOUL is fixed to 3. So, sentence boundary is introduced by using these two special tokens. Each sentence in the document is independent which means that there is no information between consecutive sentences that is taken into account. This type of SOUL model is call "standard".

Cross model If we assume that there does not exist any negligible information between sentences, we can still follow an n-gram approach by considering the whole document as one long sentence and using </s> to mark sentence boundary. The probability of the second sentence turns out to be as follows:

$$p(\text{Let's}|\text{of love </s>}).p(\text{do}|\text{love </s> Let's}).$$
$$p(\text{this}|\text{</s> Let's do}).p(\text{</s>}|\text{Let's do this}), \quad (2)$$

By doing this, we obtained the "cross" SOUL model. Theoretically, by increasing the order n, the model could take almost all words of the previous sentences into the context to predict words in the current sentence. Note that, there exists other ways to take all previous words into account, such as a "cache" maximum entropy language model [10], a recurrent neural network language model (RNNLM) [11].

Intuitively, it is evident that the information between sentences in the document is helpful. However, in practice, it is often difficult to take this type of information into account to improve the system performance, especially on large scale tasks. Conclusions for the literature for this problem are mixed at best. In [12], RNNLM was shown to work better than any other methods including n-gram NNLM. However, it is unclear that RNNLM is more efficient due to the difference in structure of the two models, or the capacity of RNNLM to take into account a long-range dependency between words (possible to be in different sentences), or both. Measuring the influence between words was once implemented in [13]. In this article, a recurrent SOUL model is shown to work only on par with a standard 10-gram SOUL models on a large scale WMT English to French translation task. The problem for this comparison is that two types of models don't have the same architecture.

For this reason, we used the same n-gram SOUL structure with large n (10) to investigate whether the words in previous sentences which have the distance to the predicted word not further than 9 is helpful in prediction.

Cross-wo-boundary model Both standard and cross SOUL models could not deal with the mismatch between training and test data. To clarify, supposing that after employing auto-segmentation, we have two phrases:

Music can be the food
of love Let's do this

The probability of the new second sentence estimated by a cross SOUL model becomes:

$$\begin{aligned} &p(\text{of}|\text{the food } \text{</s>}).p(\text{love}|\text{food } \text{</s>} \text{ of}).\\ &p(\text{Let's}|\text{<s>} \text{ of love}).p(\text{do}|\text{of love Let's}).\\ &p(\text{this}|\text{love Let's do}).p(\text{</s>}|\text{Let's do this}) \end{aligned} \quad (3)$$

Compared to Equation (2), the sentence boundary is moved two positions to the left. It leads to poor probability estimation because typically the training data do not have any sentence boundary placed in similar position.

One solution is to carry out the same auto-segmentation procedure with the acoustic training data, then using the corresponding transcriptions of the resulting audio segmentation as the training data. In this case, the training data and test data are guaranteed to be drawn from the same distribution, i.e., no mismatch exists. But now the training data does not contain "real" sentences but rather phrases. The main problem is that since the audio is required, the size of the training data for language modeling is restricted. Moreover, this solution hinders the use of out-of-domain data because there is now the mismatch between in-domain data having the associated audio from the same source as the test data and out-of-domain data often composed of "real" sentences.

Another solution is to completely ignore the sentence boundary, so the probability of the second sentence becomes:

$$\begin{aligned} &p(\text{of}|\text{be the food }).p(\text{love}|\text{the food of}).\\ &p(\text{Let's}|\text{food of love}).p(\text{do}|\text{of love Let's}).\\ &p(\text{this}|\text{love Let's do}) \end{aligned} \quad (4)$$

The underlying idea is simple: Since there is no trivial solution for detecting sentence boundary when testing, we completely ignore it in the training phase to guarantee the homogeneity between the training and test data. In equations, sentence boundary is not in the context neither in the predicted position. So we have a "cross-wo-boundary" model. It is worth noting that three types of SOUL models presented above have the same architecture. They differ only in the way of constructing the context, see Table 3 for an example about the probability of the word "of".

Table 3: *Example for three types of SOUL models*

SOUL	probability
standard	$p(\text{of}\|\text{<s> <s> <s>})$
cross	$p(\text{of}\|\text{the food </s>})$
cross-wo-boundary	$p(\text{of}\|\text{be the food})$

In Section 2.5, these three types of SOUL models will be compared experimentally in both cases where long audio signals are automatically segmented into phrases or where they are segmented manually into sentences.

2.4. Decoding Procedure

As can be seen from Figure 3, there are three main phrases constituting the decoding process. The first phase begins with the feature extraction step, followed by decoding with the baseline system (MFCC+LDA+SAT) in order to estimate the transformations for speaker adaptation (fMLLR algorithm). In the second phase, Viterbi decoding is conducted with the SGMM-bMMI model and the fMMI model separately, with fMLLR adaption using the pre-estimated transformations, resulted in one set of lattice for each system. These two lattice sets are then re-scored with the 4-gram language model. Afterward, system combination is carried out to reduce the error rates from both above systems, by exploiting lattice interpolation. In our experiment, the two systems are equally treated, by setting their lattice weights to 0.5.

The last phase is where our NNLM is applied for rescoring N-best results from the lattices. Specifically, each lattice is decoded for 1000 best outputs, in which the best output is chosen based on NNLM rescoring. To do N-best rescoring with SOUL, we follow the same scheme for RNNLM provided by Kaldi [1], i.e., we adapt related scripts for SOUL.

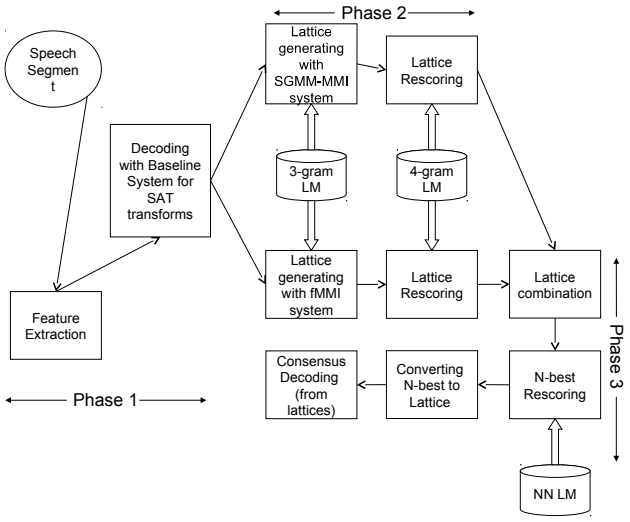


Figure 3: *Decoding diagram.*

Basically, it is done as follows. First, N-best is extracted from lattices. Then the probability estimated by SOUL models for each sentence is computed. Language model scores are updated as the interpolation of the scores provided by the back-off language model and the SOUL model. The coefficient is optimized on the development data. After that, N-best is converted back into lattices. Finally, any standard decoding method can be employed on the output lattices to have final results. In our case, consensus decoding is used at this step.

So the scripts we need to modify is for using SOUL models to compute probability for each line of a text file and combining scores of language models. For the first task, it is one of basic inference functions of standard SOUL models which can be done efficiently by using several speed-up techniques such as multi-threading, context grouping...[3]. Therefore, the computational time of N-best rescoring phase is dominated by the other steps concerning N-best extraction and lattice construction. In case of cross or cross-wo-boundary models, the computational time is similar. The only difference is that we need to use words from previous sentences while we don't have true previous sentences but their best lists. For simplification, we decide to use the best hypotheses of previous sentences provided by original lattices to predict words in a current sentence.

For the second task, it is in fact straightforward to use the script provided by Kaldi where for each sentence, a final score is the weighted average of its scores estimated by two language models. However, the interpolation in this way is only at sentence level while a (more) traditional way is to interpolate models at word level, i.e., for each word, its probability is computed as a combination of scores provided by language models. Therefore, we add scripts in order to compare these two interpolation fashions.

[3]On lattices tst2013 of $\approx$ 33 million n-grams, it costs around 4 minutes on Intel(R) Core(TM) i7-3770K CPU with Intel(R) Math Kernel Library.

2.5. Experimental results

Table 4 shows the experimental results with the three final systems. The combination technique allows us to reduce slightly the WER, by around 3% which is identical in the case of rescoring the lattice with the 4-gram language model. Besides, it is clear that auto-segmentation and speech detection exacerbated the systems' performance, by increasing the WER by 10% relatively. As mentioned above, the speech detection inevitably ignores 2% of spoken data, leading to uncompensated deletions in recognition.

Table 4: *ASR results for various acoustic models and segmentation types (manual, auto)*

System	WER			
	dev2010		tst2010	
	manual	auto	manual	auto
SGMM+MMI+4gram(1)	21.6	23.6	20.9	23.4
fMMI+4gram(2)	21.4	23.0	21.3	23.8
combine(1+2)	20.8	22.2	20.0	22.5

Table 5: *ASR results for different types of SOUL models*

System	WER			
	dev2010		tst2010	
	manual	auto	manual	auto
combine(1+2)	20.8	22.2	20.0	22.5
+ standard SOUL (inter)	18.8	20.5	18.1	20.9
+ standard SOUL	18.9	20.4	18.1	20.6
+ cross-wo-boundary SOUL	18.9	20.1	18.6	20.6
+ cross SOUL	19.0	20.4	18.4	20.8

Table 6: *Official results for English ASR task. Note that, results in tst2013 column is with auto-segmentation*

System	WER		
	tst2011	tst2012	tst2013
combine(1+2)	16.8	18.5	30.0
+ standard SOUL	14.6	16.2	27.4

In Table 5, we summarize WER results for different types of SOUL models which are used in N-best rescoring. There are some remarks drawn from these results. First, interpolation at sentence level is slightly better than at n-gram level. It supports the idea that two types of language models (back-off, SOUL) have different characteristics, so it is better to combine them at sentence level.

Second, cross model under-performs significantly standard model in both cases (manual and auto). It means that within the SOUL structure, taking into account words of previous sentences seems to be harmful rather than useful.

Third, concerning manual segmentation, the cross-wo-boundary model performed worst than the standard model. It shows that to predict a word, while words in previous sentences seems unnecessary, the role of sentence boundary is undeniable. On the contrary, in the case of auto segmentation, as the sentence boundaries for test data are not reliable, cross-wo-boundary model can potentially bring benefit. The experiments with development data showed this improvement, but unfortunately the improvement is not carried over test data. There are several possible reasons behind this phenomenon. First, we used only the best original hypotheses of the previous sentences to predict the words in the context. Second, the automatic segmentation caused the high rate of word deletion so the continuity of segmentations is not guaranteed.

Finally, all types of SOUL models bring significant improvements over the baseline system. As seen in Table 6, on all test data (tst2011, tst2012, tst2013), the standard SOUL model achieves improvements of about 10% relatively. Note that, the achievements could be more considerable if we use more than one SOUL model for N-best rescoring.

3. Machine Translation Task

In this section, we present our system used for the English to French Machine Translation task. The baseline system is based on the bilingual n-gram approach for Statistical MT [14, 15, 16]. This system is then enhanced with a SOUL language model [2]. The experimental evaluation shows that the system achieves competitive results, therefore it can be served as a baseline system for our further research.

3.1. Data setup and preprocessing

We used the training TED data provided by the campaign [17] and several datasets from the evaluation campaign of Workshop for Machine Translation (WMT) 2013[4]. We don't use Common-Crawl or any data from LDC. Considering the TED data as the in-domain data, half of the parallel dataset Giga is filtered out by applying a technique described in [7] on the French side. Note that, in our configuration, we use tst2010 as the development set and dev2010 as the test set. The reason behind this substitution is simple: We want to have more sentences in the development set than in the test set. This development data is used in the optimization procedure for the log-linear framework as well as optimizing other hyper-parameters such as the interpolation weights for language modeling, data filtering... The (internal-)test data is used to choose the best system for evaluation. The final parallel data consist of TED, NewsCommentary, Europarl and $\frac{1}{2}$ Giga. The monolingual data contain TED, News2008-2012, Europarl, Giga, UN for a total of $58{,}793{,}286$ sentences and $1{,}744{,}768{,}777$ tokens.

The preprocessing step was done as follows. As data sets are obtained from several sources, notably Internet. In order to have a clean and homogeneous data in terms of format, we decided to delete unnecessary characters, especially malform unicode ones, then converted texts into standard precomposed unicode format. We treated cases as is.

For the English side, we followed Penn Treebank style and used the script provided by Penn[5]. As we need Part-Of-Speech (POS) tags on the source side, we use TreeTagger [18] toolkit applied on the tokenized data. For the French side, the tokenization process was done by using BonSai toolkit well adapted for French[6] [19]. It separates common French phrases such as "donnez-le-nous" into three words: "donnez -le -nous". Another point of this process is that it matches compounds in a text, then replacing the space that separates the components by a "_". Compounds were taken from a built-in list, which contains phrases such as "a fortiori", "au lieu de", " partir de" ... As there is not any available scripts in Bonsai toolkit to convert tokenized texts back to original texts, we implement that task ourselves by breaking out compound words and then applying detokenization.

3.2. System overview

We used N-code to build a baseline system, hence following exactly the bilingual n-gram approach described in [14, 15, 16]. Note that, the baseline system construction is very similar to the one used in [20]. To build a translation model, word alignments were first obtained by carrying out MGIZA++[21]. Based on the information from word alignment, words in each source sentence were reordered to match the word order in its target sentence. Tuples were defined as basic translation units containing source and target phrases. Each pair of sentences was considered as a sequence of tuples. For each pair, there were maybe more than one possible sequence. Therefore, some conditions are added [15] to guarantee that there is a unique sequence of tuples which can be associated to a pair of sentences. The most important condition is that each tuple in the sequence cannot be divided into small tuples. After that, translation models are n-gram models that estimate the probability of a sequence of tuples.

When inference, translation was broken into two steps: a source reordering step and a translation step. In the source reordering step, a source sentence was represented in the form of word lattices which contains the most likely reordering hypotheses. These hypotheses were obtained by applying rewrite rules learned from word alignments and Part-of-Speech (POS) taggers of the source side. It has been shown in [16] that learning rules from POS tagger has a better generalization. In the translation step, all hypotheses in the lattice were translated monotonically using the log-linear framework.

The baseline system is the combination of four translation models based on lexicalized weighting and relative frequency (4 features), a monotone-swap-forward-backward (MSFB) lexicalized reordering model [22, 23] (8 features), a

[4] http://www.statmt.org/wmt13/translation-task.html

[5] http://www.cis.upenn.edu/ treebank/tokenizer.sed

[6] http://alpage.inria.fr/statgram/frdep/fr_stat_dep_malt.html

word bonus (1 feature), a tuple bonus (1 feature), a "weak" distance-based distortion model (1 feature), four 3-gram translation models trained on TED, NewsCommentary, Europarl and $\frac{1}{2}$ Giga (4 features) and a 4-gram language model (1 feature). It results in 20 feature functions combined in the log-linear framework. Their optimization weights are obtained by employing the MERT procedure [24]. N-gram translation models and language models are trained using SRILM toolkit with the modified interpolated Knesey-Ney smoothing technique [8].

For language modeling, the vocabulary contains 500,156 most frequent words, which occur more than 15 times in the training data. The back-off language model is the interpolation of nine 4-gram sub-models trained on each training dataset with weights optimized on the development data. The perplexity of the final model computed on the test data is 74.

We trained a 10-gram SOUL model on the same training data following exactly the procedure described in [9]. We used 500 as the dimension projection, $1000; 500$ as the size of 2 hidden layers and $2000; 2000$ as the size of the shortlist and the number of classes for out-of-shortlist words. It achieves 59 as the perplexity on the test data (20% better than the back-off model).

Proved to be helpful [25, 26], SOUL language model was used on top of this system in the 300-best rescoring phase. For each hypothesis in the list, a score of the SOUL language model is computed, then being added as a new score. Weights of models are re-optimized following the MERT procedure on the development data.

3.3. Experimental results

MT systems are evaluated according to BLEU, NIST metrices computed by the script provided by NIST[7]. The results are summarized in Tables 7 and 8. Note that the results in Table 7 are computed on tokenized texts while the results in Table 8 are the official results provided by organizer. We see that 300-best rescoring with SOUL improves significantly the performance (≈ 1.4 BLEU points improvement).

Table 7: *Results for English to French MT task. Scores are case-sensitive with tokenized texts*

Systems	Scores			
	dev data		test data	
	BLEU	NIST	BLEU	NIST
baseline	34.9	7.55	28.6	6.60
+ rescoring with SOUL	35.8	7.69	29.7	6.73

4. Conclusion

In this paper, our systems served for the English ASR and English to French MT tasks of IWSLT2013 were presented in detail. On the ASR task, by comparing three types of SOUL language models distinguished in the way of treating sentence boundary, we found that in the SOUL structure, taking into account words of previous sentences were not effective, even in the case of auto-segmentation. In both tasks, the SOUL models were used on top in N-best rescoring phase. They were proved to improve significantly the system performance with approximately 10% relative WER reduction for ASR task and an addition of about 1.4 BLEU points for MT task.

Table 8: *Official results for English to French MT task. BLEU is case-sensitive*

Systems	BLEU		
	tst2011	tst2012	tst2013
baseline	37.1	38.6	36.2
+ rescoring with SOUL	38.8	39.9	37.6

This work was partially supported by National ICT Project KC.01.03/11-15

5. References

[1] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[2] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured Output Layer neural network language model," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, may 2011, pp. 5524 –5527.

[3] A. Katsamanis, M. Black, P. G. Georgiou, L. Goldstein, and S. S. Narayanan, "SailAlign: Robust long speech-text alignment," in *Proc. of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, Jan. 2011.

[4] H. Yamamoto, Y. Wu, C.-L. Huang, X. Lu, P. R. Dixon, S. Matsuda, C. Hori, and H. Kashioka, "The NICT ASR system for IWSLT 2012," in *International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 12 2012.

[5] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafit, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas, "The subspace Gaussian mixture model - A structured model for speech recognition," *Computer Speech & Language*, vol. 25, no. 2, pp. 404 – 439, 2011.

[6] S. Meignier and T. Merlin, "LIUM SpkDiarization: an

[7] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a.pl

open source toolkit for diarization," in *CMU SPUD Workshop*, Dallas (Texas, USA), mars 2010.

[7] R. C. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *Proceedings of the ACL 2010 Conference Short Papers*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 220–224.

[8] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Technical Report TR-10-98*. Cambridge, Massachusetts, USA: Computer Science Group, Harvard University, 1998.

[9] H.-S. Le, I. Oparin, A. Messaoudi, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Large Vocabulary SOUL Neural Network Language Models," in *Proceedings of the 12th Annual Conference of the INTERSPEECH 2011*, Florence, Italy, 2011.

[10] R. Rosenfeld, "Adaptive Statistical Language Modeling: A Maximum Entropy Approach," Ph.D. dissertation, Carnegie Mellon University, 1994.

[11] T. Mikolov, M. Karafit, L. Burget, J. ernock, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the 11th Annual Conference of the INTERSPEECH 2010*, vol. 2010, no. 9, 2010, pp. 1045–1048.

[12] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. Cernocký, "Empirical Evaluation and Combination of Advanced Language Modeling Techniques," in *Proceedings of the 12th Annual Conference of the INTERSPEECH 2011*, 2011, pp. 605–608.

[13] H.-S. Le, A. Allauzen, and F. Yvon, "Measuring the Influence of Long Range Dependencies with Neural Network Language Models," in *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 1–10.

[14] F. Casacuberta and E. Vidal, "Machine Translation with Inferred Stochastic Finite-State Transducers," *Comput. Linguist.*, vol. 30, no. 2, pp. 205–225, June 2004.

[15] J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Costa-jussà, "N-gram-based Machine Translation," *Comput. Linguist.*, vol. 32, no. 4, pp. 527–549, Dec. 2006.

[16] J. M. Crego and J. B. Mariño, "Improving statistical MT by coupling reordering and decoding," *Machine Translation*, vol. 20, no. 3, pp. 199–215, Sept. 2006.

[17] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[18] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *Proceedings of International Conference on New Methods in Language Processing*, 1994.

[19] M. Candito, J. Nivre, P. Denis, and E. Henestroza Anguiano, "Benchmarking of Statistical Dependency Parsers for French," in *Coling 2010: Posters*. Beijing, China: Coling 2010 Organizing Committee, August 2010, pp. 108–116.

[20] A. Allauzen, N. Pcheux, Q. K. Do, M. Dinarelli, T. Lavergne, A. Max, H.-S. Le, and F. Yvon, "LIMSI @ WMT13," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August 2013, pp. 62–69.

[21] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, ser. SETQA-NLP '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 49–57.

[22] C. Tillmann, "A unigram orientation model for statistical machine translation," in *Proceedings of HLT-NAACL 2004: Short Papers*, ser. HLT-NAACL-Short '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004, pp. 101–104.

[23] J. M. Crego, F. Yvon, and J. B. Mario, "N-code: an open-source Bilingual N-gram SMT Toolkit," *Prague Bulletin of Mathematical Linguistics*, vol. 96, pp. 49–58, 2011.

[24] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ser. ACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 160–167.

[25] T. Lavergne, A. Allauzen, H.-S. Le, and F. Yvon, "LIMSIs experiments in domain adaptation for IWSLT11," in *Proceedings of the 8th International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, 2011.

[26] A. Allauzen, H. Bonneau-Maynard, H.-S. Le, A. Max, G. Wisniewski, F. Yvon, G. Adda, J. M. Crego, A. Lardilleux, T. Lavergne, and A. Sokolov, "LIMSI @ WMT11," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, July 2011, pp. 309–315.

TÜBİTAK TURKISH-ENGLISH SUBMISSIONS for IWSLT 2013

Ertuğrul Yılmaz, İlknur Durgar El-Kahlout, Burak Aydın,
Zişan Sıla Özil, Coşkun Mermer

TÜBİTAK-BİLGEM
Gebze 41470, KOCAELİ, TURKEY
{yilmaz.ertugrul,ilknur.durgar,burak.aydin,zisan.ozil,coskun.mermer}@tubitak.gov.tr

Abstract

This paper describes the TÜBİTAK Turkish-English submissions in both directions for the IWSLT'13 Evaluation Campaign TED Machine Translation (MT) track. We develop both phrase-based and hierarchical phrase-based statistical machine translation (SMT) systems based on Turkish word- and morpheme-level representations. We augment training data with content words extracted from itself and experiment with reverse word order for source languages. For the Turkish-to-English direction, we use Gigaword corpus as an additional language model with the training data. For the English-to-Turkish direction, we implemented a wide coverage Turkish word generator to generate words from the stem and morpheme sequences. Finally, we perform system combination of the different systems produced with different word alignments.

1. Introduction

We participated in the IWSLT Evaluation Campaign for the Turkish-English MT track for both directions. The typological, morphological and word order differences of this language pair make the implementation of SMT systems a challenging task. English and Turkish are typologically rather distant languages. English has a very limited morphology and rather fixed Subject-Verb-Object (SVO) constituent order. However, Turkish is an agglutinative language with very flexible (but Subject-Object-Verb (SOV) dominant) constituent order, and has a very rich and productive derivational and inflectional morphology where word structures might correspond to complete phrases of several words in English when translated.

Overview of the systems can be summarized as follows: (1) We used the feature-based representation of Turkish in order to aggregate the statistics from different morphological forms of the words in addition to the word representation as in [1], (2) We compared phrase-based SMT systems with hierarchical phrase-based systems, (3) We augmented the training data with the content words extracted from itself to bias the stem word alignments as in [2], (4) We used reverse word order of the source language in order to obtain alternative translations similar to [3], (5) We preferred to use WIT corpus for the translation model training, (6) We combined both SETIMES and WIT corpora as one language model for English-to-Turkish systems, (7) We implemented a wide-coverage Turkish morphological word generator to generate Turkish words from stem and morpheme sequences, (8) We added Gigaword corpus as an additional language model for Turkish-to-English systems, (9) We combined systems (2), (3) and (4) to improve the translation quality from different word alignments. As a result, we improved $+1.4$ BLEU points on *test2011* and $+1.5$ BLEU points on *test2012* compared to the best system of IWSLT'12 Turkish-to-English MT track.

This paper is organized as follows: Section 2 introduces the challenges of implementation of SMT systems for the Turkish-English language pair and summarizes the previous work. Section 3 describes the data sets and explains the experimental setups. Section 4 shows the experimental results in both directions and reports the official submission scores. We conclude with Section 5.

2. Turkish-English Statistical Machine Translation

Turkish exhibits interesting properties from an SMT point of view. Its agglutinative structure has very productive inflectional and derivational processes for word formation. Words are created by concatenating morphemes to the stem word or to other morphemes. Generally, word formation is done by suffixation. Except for very few cases, surface realizations of the morphemes are conditioned by various regular morphophonemic processes such as vowel harmony, consonant assimilation, and elisions. The morphotactics of word forms could be quite complex when multiple derivations are involved. The average number of bound morphemes (i.e., excluding the stem) in words is about two. The productive morphology of Turkish potentially implies a very large vocabulary size. In most cases, single Turkish words typically tend to align with whole phrases on the English side when sentence pairs are aligned at the word level.

During the development of SMT systems, morphological preprocessing is useful and sometimes crucial when at least one of the languages is morphologically complex. Turkish is one of the languages that need special attention as

several derivational and inflectional processes can produce very complex Turkish words. Mapping the rich morphology of Turkish to the limited morphology of English has been addressed by several researchers. To reduce the large vocabulary size and to force more one-to-one word alignments, researchers prefer a sub-word representation of the foreign language while translating to/from English. The research showed that replacing the Turkish word representation with a sub-word representation performs better in the translation process in both directions. [1, 2, 4] used morphological analysis to separate some Turkish inflectional morphemes that have counterparts on the English side in English-to-Turkish SMT. Along the same direction, [5] applied syntactic transformations such as joining function words on the English side to the related content words. [6] used an unsupervised learning algorithm to find the segmentations automatically from parallel data. [7] presented a series of segmentation schemes to explore the optimal segmentation for statistical machine translation of Turkish. In addition, an important amount of effort was spent by several research groups on Turkish-to-English SMT in the IWSLT'09[1] BTEC task, IWSLT'10[2] and IWSLT'12[3] TED tasks.

3. Experiments

In the experiments, we used all supplied monolingual and parallel texts for the system development. We tuned systems with *dev2010* and used *test2010* as the internal test set. In terms of official evaluation of the translation systems, we submitted the last two years' test sets *test2011*, *test2012* and a new test set *test2013*. As we noticed that some portions of the Turkish texts in WIT corpus [8] are asciified, we employed a deasciifier tool[4] to clean these portions of the data.

Tables 1 and 2 show the Turkish (with word and morpheme-based representations) and English statistics after the pre-cleaning step. One can notice that with full segmentation, the number of unique words of Turkish and English are closer than the word representation. As the vocabulary sizes of the languages get closer, we expect better word alignments.

Table 1: WIT training data statistics

	Sentences	Unique words	Total words
Turkish (Word)	130K	158K	1.8M
Turkish (Feature)		35K	2.9M
English		45K	2.5M

[1] www2.nict.go.jp/univ-com/multi_trans/WS/IWSLT2009
[2] iwslt2010.fbk.eu
[3] hltc.cs.ust.hk/iwslt
[4] turkce-karakter.appspot.com

Table 2: SETIMES training data statistics

	Sentences	Unique words	Total words
Turkish (Word)	173K	143K	3.9M
Turkish (Feature)		43K	5.5M
English		60K	4.6M

3.1. Phrase-based vs. Hierarchical Phrase-based Systems

Although phrase-to-phrase translation [9] overcomes many problems of word-to-word translation [10] and has been successful for some language pairs during the last decade, the continuity of phrases is its main shortcoming. Clearly, this is a problem for language pairs with very different word orders such as Chinese-English. For that kind of language pairs, to generate the target phrase, we may need sub-phrases from different parts of the source sentence which are distant from each other. To overcome the limitations of the phrase-based model, Chiang [11] has introduced a hierarchical phrase-based model that uses bilingual phrase pairs to generate hierarchical phrases that allow gaps and enable longer distance reorderings. Previous work [1, 7] showed that using hierarchical phrase-based (HPB) decoder outperforms the phrase-based (PB) systems for Turkish-English.

For this reason, we performed experiments mainly with HPB decoders but also implemented systems with PB decoders in order to use the output of the PB systems in the internal system combination.

3.2. Sub-word Representation

We implemented the baseline experiments with the word-level representation of Turkish. As mentioned in Section 2, incorporating morphology when working with morphologically rich(er) languages in SMT performs better than the word-level. For this reason, in the further experiments, we preferred using a feature-based representation of Turkish in both directions as this representation dramatically reduces the vocabulary size on the Turkish side as shown in Tables 1 and 2. To produce the feature-based word representation, we first pass each word through a morphological analyzer [12]. The output of the analyzer contains the morphological features encoded for all possible analyses and interpretations of the word. Then we perform morphological disambiguation on the morphological analyses [13]. Once the contextually-salient morphological interpretation is selected, we remove the redundant morphological features that do not correspond to a surface morpheme such as part-of-speech features *Noun*, *Verb etc.*, 3rd singular agreement feature *A3sg*, and positiveness feature *Pos* and so on. There only remain features that correspond to lexical morphemes making up a word such as dative *Dat*, accusative *Acc*, past participle *PastPart* and so on.

We segmented the morphologically-analyzed Turkish

sentences at every feature boundary, denoted by the (_) symbol. A typical sentence pair with Turkish word representation and full segmentation is as follows:

- **Word representation:** Organize edeceğim , yöneteceğim ve onu dünyaya sunacağım .
- **Full segmentation:** Organize et _Fut _A1sg , yönet _Fut _A1sg ve o _Acc dünya _Dat sun _Fut _A1sg .
- **Reference:** I'm going to organize it and direct it and get it going in the world .

3.3. Content Words

From the morphologically segmented corpora, we also extract for each sentence in the training corpus, the sequence of stems for open-class content words (Nouns, Adjectives, Adverbs, and Verbs). For Turkish, this corresponds to removing *all* morphemes and any stems for closed classes.

For English, we used the TreeTagger [14] to tag the sentences and removed all words tagged as closed class words along with the tags such as *+VVG* that signal a morpheme on an open-class content word. We use this data to augment the training corpus and bias content word alignments, with the hope that such stems may get a better chance to align without any additional "noise" from morphemes and other function words. An example of a content word (bold) sentence pair of is as follows:

- **Turkish content words: Organize et** _Fut _A1sg , **yönet** _Fut _A1sg ve o _Acc **dünya** _Dat **sun** _Fut _A1sg .
- **English content words:** I +PP am +VB **go** +VVG to +TO **organize** +VV it +PP and +CC **direct** +VV it +PP and +CC **get** +VV it +PP **go** +VVG in +IN the +DT **world** +NN . +SENT

Table 3 shows the Turkish and English content word corpus statistics after the pre-cleaning step.

Table 3: WIT content word statistics

	Sentences	Unique words	Total words
Turkish	128K	45K	1.1M
English	128K	39K	1M

3.4. Reverse Translation

Word order differences affect many steps of the translation process such as word alignment, phrase extraction, and thus the translation quality. It has been observed that one gets better alignments and hence better translation results when the word orders of the source and target languages are more or less the same. When word orders are different, it can be useful to systematically reorder the tokens of source sentences to an order matching or very close to the target language word order so that alignments could be very close to a monotonic one. Thus instead of forcing the decoders to employ reordering schemes, the source sentences are similarly reordered and then decoded with the decoder employing more simple reordering models. As the word orders of Turkish (SOV) and English (SVO) differ, reordering of the source sentence may allow to produce an alternative translation table thus alternative translation performance. In order to make the word orders especially *Verbs* a bit closer, one approach can be to use the reverse order of the source side of the language pair. In these experiments, we reversed the order of the source language similar to [3] before the word alignment step as generally reordering target language is not preferred because of the need of an additional post-processing. Reverse sentence examples of the source language for two translation directions are as follows:

- **Turkish reverse sentence:** . _A1sg _Fut sun _Dat dünya _Acc o ve _A1sg _Fut yönet , _A1sg _Fut et Organize
- **English reverse sentence:** . world the in going it get and it direct and it organize to going I'm

4. Results

For the IWSLT'13 Evaluation Campaign, we performed several SMT experiments for Turkish-English with different settings. All available data was tokenized with an in-house Turkish tokenizer and then truecased. We generated word alignments using MGIZA [15] with default settings. We implemented both the phrase-based and the hierarchical phrase-based systems with Moses Open Source Toolkit [16]. The system parameters were optimized with the minimum error rate training (MERT) algorithm [17] on the tune set *dev2010*, evaluated on the test set *test2010*, and scores are reported in terms of BLEU [18]. We trained conventional 5-gram language models (LMs) from the parallel corpus for both directions but also performed tests with 4-gram Gigaword language model for the Turkish-to-English systems. All language models were trained with the SRILM toolkit [19] using modified Kneser-Ney smoothing [20] and then binarized using Kenlm [21] .

For phrase-based systems, we allowed unlimited jumps for word reordering (*distortion-limit* $= -1$). At each step, systems were tuned with five different seeds with lattice-samples and minimum Bayes risk decoding; *-mbr* [22] is employed during the decoding.

For hierarchical phrase-based systems, we relaxed the rule table extraction by allowing sub-phrases of any size to be replaced by a non-terminal(*-MinHoleSource* $= -1$), and set *-cube-pruning-pop-limit* to 5000 to increase the number of hypotheses created in each span.

4.1. Turkish-to-English

The baseline experiment was conducted with the hierarchical phrase-based system and Turkish word representation (Exp. #1), then we employed the morpheme-based representation as explained in Section 3.2 which results in an improvement of $+2.5$ BLEU points (Exp. #2). We experimented to remove the out-of-domain data *SETIMES* corpus from the training that gave us a $+1.1$ BLEU point increase (Exp. #3). Further, including the 4-gram Gigaword corpus as an additional language model improved the performance of the system by 1.1 BLEU points (Exp. #4). We performed two more experiments with augmenting the corpus with content words (Exp. #5) and using the reverse word order on the source side (Exp. #6) which resulted in a -0.4 and -1.0 BLEU points decrease, respectively. We also repeated the experiments 4, 5, and 6 with the phrase-based framework.

Table 4: Turkish-to-English BLEU scores

System	dev2010	test2010
1. HPB - Word Rep.	11.31	12.47
2. HPB - Feature Rep.	13.54	14.96
3. 2 + WIT only	14.00	16.10
4. 3 + Gigaword	15.33	17.14
5. 4 + Content Corpus	14.80	16.68
6. HPB Reverse Corpus	14.18	16.18
7. 4 with PB	13.22	15.69
8. 5 with PB	13.53	15.95
9. 6 with PB	13.00	14.77

Table 4 shows the experimental results on the development and test sets. All of the experiments run with five tuning seeds and the one with the maximum score is selected after each step. We observed that the reported improvements are consistent in all tuning runs[5]. Although not reported here, using Turkish-specific tokenizer improved the performance by $+0.3$ BLEU points. As expected, the HPB systems outperform the PB systems by approximately 1.5 BLEU points. Adding content word corpus degraded the performance in the HPB framework but induced a slight improvement (0.3 BLEU points) in the PB systems. Experiments showed that using out-of-domain data without performing any domain adaptation method hurts the performance of the systems. Reverse word order in the source language is slightly worse than the exact word order individually but this system can be used as a candidate in the system combination which will be explained later. We also performed experiments with combined language model where SETIMES and WIT corpora are concatenated and trained together but observed a decrease of 0.2 BLEU points.

[5]Variation between tunes are approximately 0.4 BLEU points

4.2. English-to-Turkish

In this case, the target language is the morphologically-complex Turkish. This presents a challenge in predicting the correct word forms (or their morphological composition) using a sparser target language model data. In the morpheme-based system, there is a need for a word-generation tool that generates Turkish words from stem and morpheme sequences. The performance of this tool will directly affect the translation quality of the morpheme-based system. The challenge in generating Turkish word forms is that Turkish word features can be mapped to several suffixes and each combination leads to a different Turkish word. Moreover, during the generation process the vowel harmony should be taken into consideration.

Most of the experiments of Section 4.1 were repeated for the English-to-Turkish direction. Similar to the Turkish-to-English experiments, the baseline experiment was conducted with the hierarchical phrase-based system using Turkish word representation (Exp. #1), then we experimented with Turkish morpheme-based representation which results in an improvement of $+0.6$ BLEU points (Exp. #2). We also removed the out-of-domain data *SETIMES* corpus from the training, which resulted in an increase of $+0.1$ BLEU points (Exp. #3). We performed experiments with the combined language model which induced a $+0.1$ BLEU improvement (Exp. #4). Above that, we performed experiments by augmenting the corpus with content words (Exp. #5) and using the reverse word order on the source side (Exp. #6) which resulted in a -0.3 and -0.4 BLEU points decrease, respectively. Again, we also repeated the experiments 4, 5, and 6 with the phrase-based framework.

Table 5: English-to-Turkish BLEU scores

System	dev2010	test2010
1. HPB - Word Rep.	6.11	7.70
2. HPB - Feature Rep.	7.14	8.31
3. 2 + WIT only	6.34	8.41
4. 3 + Combined LM	6.07	8.52
5. 3 + Content Corpus	6.54	8.24
6. HPB Reverse Corpus	5.99	8.13
7. 4 with PB	4.91	7.40
8. 5 with PB	4.91	7.23
9. 6 with PB	4.32	6.83

Table 5 shows the experimental results on the development and test set. Similar to Turkish-to-English direction, the HPB systems outperform the PB systems by approximately 1.1 BLEU points. Adding content word corpus and reverse word order hurts the performance in both HPB and PB systems but they were kept for the system combination. Employing combined language model increased the system performance in the test set contrary to the Turkish-to-English experiments.

Table 6: The word statistics of morphological generation for outputs of Exp. #4. (#stems: words with no morphemes, hence no word generation is required, #sequences: words of the form stem+morphemes, found: sequence words for which an exact single-word-form is found; sub-found: sequence words resolved after elimination of some trailing morphemes)

	total	**#stems**	**#sequences**	**found (%)**	**sub-found (%)**	**missed (%)**
test2010	23056	13604	9452	9065 (95.9%)	167 (1.8%)	220 (2.3%)
test2011	19447	11312	8135	7793 (95.8%)	124 (1.5%)	218 (2.7%)
test2012	22021	12609	9352	8878 (94.9%)	174 (1.9%)	300 (3.2%)
test2013	16410	9414	6996	6643 (95.9%)	132 (1.9%)	221 (3.2%)

4.2.1. Turkish Word Generation

In morpheme-based translation, a word generation tool is required to generate the correct Turkish word from the outputs of systems which contain words represented with stems and sequence of morphemes. We used an in-house morphological generation tool that, given a text with words in a format where each morpheme is concatenated to the previous morpheme or stem, transforms these representations to the correct single-word form. This generation tool has been trained by a large Turkish corpus and works by simply creating a reverse-map through morphological segmentation of the corpus. This map contains stem+morpheme sequences as keys and their corresponding single-word forms as values. While creating this map, the disambiguation step of morphological segmentation is omitted to increase the coverage, as keeping multiple resolutions for a single-word form increases the number of keys for the reverse-map. An additional map is generated through morphological segmentation of WIT and SETIMES corpora to further increase coverage. These two maps are combined giving the preference to the latter map in case of disagreements.

The following are the working steps of the generation tool:

1. The system outputs and the combined map of 'stem+morphemes to single-word form' is taken.
2. Iterating through tokens, if an encountered token is:
   (a) a stem; simply output the token.
   (b) a 'stem+morphemes' that is in the map; output its value.
   (c) otherwise; drop the trailing morpheme, and go to 2a.

Examples of word generation are as follows:

- **Stem+Morpheme Sequence:** et_Aor_A1sg
  Surface Form: ederim[6]
- **Stem+Morpheme Sequence:** duy_PastPart_P3sg
  Surface Form: duydu[7]

[6] I do it
[7] he/she/it heard

Step 2c in this procedure can help in cases where an extraneous morpheme is found at the end of a word, which in turn would increase the coverage of the generator. Table 6 shows the coverage of the word generator for outputs of (Exp. #4) for all the test data. For about 95% of the tokens of the form stem+morpheme sequences, the procedure finds an exact single-word form match. An additional 1-2% match is achieved by following the process of dropping the trailing morpheme and re-checking the map for the resulting sequence. For 2% to 3% of the words of the form stem+morphemes, all morphemes are eliminated and only the stem is represented in the output (missed).

4.3. System Combination

System combination attempts to improve the quality of machine translation output by combining the outputs of different translation systems which usually are based on different paradigms such as phrase-based, hierarchical, etc. aiming to exploit and combine strengths of each system. The outputs of some of our translation systems, which are based on different methods as explained in the previous sections, were put into a combination task. We combined the outputs of some of the best performing -best tuning run in terms of BLEU score- hierarchical phrase-based systems using the open-source system combination tool, MEMT [23]. We also experimented with adding phrase-based systems to the combination task but did not observe any improvements, hence we provide results for combination of different hierachical phrase-based systems.

MEMT should ideally be tuned by a separate held-out data that is different from system training and tuning data. As we did not have additional tuning data for system combination tuning, we primarily used *dev2010* data to tune the system combination decoder. To see how having separate tuning data for system combination would have effected the quality of the combined system outputs, we trained the system combination decoder with *test2010* data evaluating the performance on *test2011*, *test2012*, and *test2013* data (not tested for *test2010* as it would not be valid). Tuning the system combination decoder with *test2010* data yielded comparable results with the system tuned by *dev2010* data. Also, tuning with the combination of *dev2010* and *test2010* data yielded similar results. The results we provide in this paper are for system combination tasks that employed either only

dev2010, or both *dev2010* and *test2010* data as tuning data.

The language models used for system combination training and decoding were i) a 4-gram language model constructed from the Gigaword database for Turkish-English translations, and ii) a 5-gram language model constructed from the combination of WIT and SETIMES corpora for English-Turkish translations.

Table 7: BLEU scores of individual systems and their system combinations for English-to-Turkish. (*Exp. #3 with different tuning seed)

Experiment	test2010	test2011	test2012	test2013
3*	8.84	8.85	8.81	8.50
4	8.52	8.86	9.20	8.65
5	8.24	8.74	8.70	8.08
Sys. Combo.	8.82	**9.16**	**9.29**	**8.97**
6	8.13	7.99	8.57	8.05
Sys. Combo.	8.77	**9.34**	**9.48**	**8.86**

Table 7 shows the BLEU scores of some individual systems as well as the the BLEU score of their combined outputs for English-to-Turkish translations. Combining the outputs of experiments 3, 4, and 5 yields about the same BLEU score for *test2010* and better BLEU scores for test sets 2011, 2012, and 2013 compared to the best individual system. Combination of the outputs of those three systems achieves a relative BLEU improvement of about 3.5%, 0.98%, and 3.7% over the best performing individual systems for test sets 2011, 2012, and 2013, respectively. Integration of a fourth system, experiment 6, to the combination task provides better improvements for *test2011*(5.5%) and *test2012*(3.0%) data, but yields a lower score for *test2010* and *test2013* data compared to the combination of 3, 4, and 5. The official results we submitted to IWSLT'13 are the combined outputs of systems 3, 4, and 5. For the submitted combined outputs, the improvements over the best performing individual systems for *test2011* and *test2013* were computed to be statistically significant ($p<0.05$).

Table 8 shows the BLEU scores of individual systems and combined systems for the opposite translation direction, Turkish to English. Using only *dev2010* data for system combination decoder tuning (Sys. Combo. (dev only)), the combined system outputs in this direction provided about 1.17% improvements for both *test2010* and *test2012* over the best performing individual systems, and no improvements for *test2011* and *test2013* data. Adding *test2010* data into the tuning of the system combination decoder (Sys. Combo. (dev+test)) provided some improvement for *test2011* and *test2013* over (Sys. Combo (dev only)). The combined systems -compared to the best individual system- provided statistically significant improvements for this translation direction for *test2010* and *test2012* data ($p<0.05$), and performed worse or about the same for *test2011* and *test2013* data.

Our official submissions for English-to-Turkish and Turkish-to-English are the fourth rows of Tables 7 and 8.

5. Conclusions

This paper presented the experiments and results of the TÜBİTAK Turkish-English submissions in both directions for the IWSLT'13 Evaluation Campaign TED Machine Translation (MT) track. Due to the rich morphological and syntactic properties of Turkish, statistical machine translation involving Turkish implies processes that are more complex than standard statistical translation models.

In our implemented systems, we improved from 12.47 to 17.34 BLEU points in Turkish-to-English SMT systems and 7.70 to 8.82 in English-to-Turkish SMT systems on the *test2010* set. For Turkish-to-English, we improved $+1.4$ BLEU points on *test2011* and $+1.5$ BLEU points on *test2012* compared to the best system of IWSLT'12 Turkish-to-English MT track. Major results of our work can be summarized as follows:

- We compared the feature-based and word representation of Turkish,
- We compared phrase-based SMT systems with hierarchical phrase-based systems,
- We augmented the training data with the content words extracted from itself,
- We used reverse word order of the source language in order to obtain alternative translations,
- We preferred to use WIT corpus for the training,
- We added Gigaword corpus as an additional language model for Turkish-to-English systems,
- We combined both SETIMES and WIT corpora as one language model for English-to-Turkish systems,
- We implemented a wide-coverage Turkish morphological word generator to generate Turkish words from stem and morpheme sequences,
- We applied system combination to hierarchical phrase-based systems that are trained on different representations of the training corpora.

6. References

[1] I. D. El-Kahlout, C. Mermer, and M. U. Doğan, "Recent Improvements In Statistical Machine Translation Between Turkish and English," in *Multilingual Processing in Eastern and Southern EU Languages.- Low-resourced Technologies and Translation*. Cambridge: Cambridge Scholars Publishing, 2012. ??-??. Print.

[2] I. Durgar El-Kahlout and K. Oflazer, "Exploiting morphology and local word reordering in English-to-Turkish phrase-based statistical machine translation,"

Table 8: BLEU scores of individual systems and their system combinations for Turkish-to-English. (*Exp. #5 with different tuning seed)

Experiment	test2010	test2011	test2012	test2013
4	17.14	18.77	18.62	18.88
5*	16.59	18.29	18.53	18.40
6	16.18	17.76	17.61	17.59
Sys. Combo. (dev only)	**17.34**	18.63	**18.93**	18.67
Sys. Combo. (dev+test)	N/A	**18.83**	**18.84**	18.70

IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, no. 6, pp. 1313–1322, 2010.

[3] M. Huck, S. Peitz, M. Freitag, M. Nuhn, and H. Ney, "The RWTH Aachen machine translation system for WMT 2012," in *Proceedings of the Seventh Workshop on Statistical Machine Translation (WMT)*, 2012, pp. 304–311.

[4] K. Oflazer, "Statistical machine translation into a morphologically complex language," in *Computational Linguistics and Intelligent Text Processing, 9th International Conference, CICLing 2008, Haifa, Israel*, ser. Lecture Notes in Computer Science, vol. 4919, 2008, pp. 376–387.

[5] R. Yeniterzi and K. Oflazer, "Syntax-to-morphology mapping in factored phrase-based statistical machine translation from English to Turkish," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ser. ACL '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 454–464. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858681.1858728

[6] C. Mermer and A. A. Akin, "Unsupervised search for the optimal segmentation for statistical machine translation," in *Proceedings of the ACL 2010 Student Research Workshop*, ser. ACLstudent '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 31–36. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858913.1858919

[7] N. Ruiz, A. Bisazza, R. Cattoni, and M. Federico, "FBK's machine translation systems for IWSLT 2012's TED lectures," in *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, 2012, pp. 61–68.

[8] M. Cettolo, C. Girardi, and M. Federico, "WIT3: Web inventory of transcribed and translated talks," in *Proceedings of EAMT*, 2012, pp. 261–268.

[9] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, 2003, pp. 127–133.

[10] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, pp. 263–311, 1993.

[11] D. Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[12] K. Oflazer, "Two-level description of Turkish morphology," *Literary and Linguistic Computing*, vol. 9, pp. 137–148, 1994.

[13] H. Sak, T. Güngör, and M. Saraçlar, "Morphological disambiguation of Turkish text with perception algorithm," in *Proceeding of CICLING, LNCS 4394*, 2007, pp. 107–118.

[14] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proceedings of the International Conference on New Methods in Language Processing*, 1994.

[15] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Proceedings of ACL WSETQANLP*, 2008, pp. 49–57.

[16] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of ACL Demo and Poster Session*, 2007, pp. 177–180.

[17] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2003, pp. 160–167.

[18] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, pp. 311–318.

[19] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2002, pp. 901–904.

[20] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1995, pp. 181–184.

[21] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 2011, pp. 187–197.

[22] S. Kumar and W. Byrne, "Minimum Bayes-risk decoding for statistical machine translation," in *Proceedings of HLT-NAACL*, 2004, pp. 169–176.

[23] K. Heafield and A. Lavie, "Combining machine translation output with open source: The Carnegie Mellon multi-engine machine translation scheme," *The Prague Bulletin of Mathematical Linguistics*, vol. 93, pp. 27–36, 2010.

FBK's Machine Translation Systems for the IWSLT 2013 Evaluation Campaign

Nicola Bertoldi[1], *M. Amin Farajian*[1,2], *Prashant Mathur*[1,2], *Nicholas Ruiz*[1,2], *Marcello Federico*[1]

[1]Fondazione Bruno Kessler
HLT Unit
Via Sommarive, 18
38123 Trento, TN, Italy

[2]University of Trento
ICT Doctoral School
Via Sommarive, 5
38123 Trento, TN, Italy

Abstract

This paper describes the systems submitted by FBK for the MT track of IWSLT 2013. We participated in the English-French as well as the bidirectional Persian-English translation tasks. We report substantial improvements in our English-French systems over last year's baselines, largely due to improved techniques of combining translation and language models. For our Persian-English and English-Persian systems, we observe substantive improvements over baselines submitted by the workshop organizers, due to enhanced language-specific text normalization and the creation of a large monolingual news corpus in Persian.

1. Introduction

FBK's machine translation activities in the IWSLT 2013 Evaluation Campaign [1] focused on the speech recognition and translation of TED Talks[1], a collection of public speeches on a variety of topics and with transcriptions available in multiple languages. In this paper, we describe our participation in the Machine Translation translation tasks in the official English-French as well as the optional English-Persian and Persian-English languages. These tasks entail translating subtitles transcribed and translated by the TED community.

We begin with an overview of the domain adaptation techniques used by each of our language pair experiments in Section 2: namely, data filtering and methods to combine translation models, reordering models, and language models from multiple corpora, respectively. In Section 3, we describe several experiments in the English-French translation task. In Section 4, we describe our first efforts at translated to and from English and Persian, a language pair with few parallel resources available. We introduce our efforts to collect and preprocess Perian corpora to improve the quality of Persian translation and show significant improvements over the state of the art. In Section 5 we summarize our findings.

For all language pairs, we set up a standard phrase-based system using the Moses toolkit [2]. We construct a statistical log-linear models including domain-adapted phrase translation and hierarchical reordering models [3, 4, 5], one or more target language models (LM), as well as distortion, word, and phrase penalties.

[1]http://www.ted.com/talks

2. Domain adaptation techniques

In this section, we summarize several well-known techniques for domain adaptation we applied to build high-performance models for our SMT submissions.

2.1. Data selection

The idea of data selection is to find the subset of sentences within an out-of-domain corpus that better fits with a given in-domain corpus.

To this purpose, we follow the procedure described in [6], which adapts the cross-entropy difference scoring technique introduced by [7] toward bitext data selection. First, all sentence pairs of the out-of-domain corpus are associated with a source- and target-side scores, each computed as the basic technique proposes for the corresponding monolingual scenarios, using the in-domain (TED) data as a seed and LMs of order 3. Then, the sentences are sorted according to the sum of these two scores. Finally, the optimal split between useful and useless sentences is found by minimizing the source-side perplexity of a development set on growing percentages of the sorted corpus. In our experiments, dev2010 and tst2010 are concatenated and used as the filtering development set.

2.2. Translation model combination

Three methods are applied in our submissions to combine the TM built on the available parallel training corpora: namely, fill-up [8, 9], back-off, and interpolation.

2.2.1. Fill-up and Back-off

In the fill-up approach, out-of-domain phrase pairs that do not appear in an in-domain (TED) phrase table are added, along with their scores – effectively filling the in-domain table with additional phrase translation options. The fill-up process is performed in a cascaded order, first filling in miss-

ing phrases from the corpora that are closest in domain to TED. Moreover, out-of-domain phrase pairs with more than four source tokens are pruned.

Following [8, 9] the fill-up approach adds k-1 provenance binary features to weight the importance of out-of-domain data, where k is the number of phrase tables to combine. A similar back-off approach performs the fill-up technique, but does not add any provenance binary features.

2.2.2. Linear interpolation

A common approach for building multi-model is through the linear interpolation of component models. Various approaches have been suggested for computing the coefficients of the interpolated model, the most recent being perplexity minimization described in [10] where the perplexity of each component translation model is minimized on the parallel development set. However, the mixing coefficients can be separately computed by several other techniques. In this paper, instead of calculating translation model perplexity we calculate language model perplexity on target side development set. After minimizing perplexity we get the interpolation weights which we then use as mixing coefficients for component translation models.

2.3. Reordering model combination

All techniques available for combining the TMs can be applied straightfowardly to combine the RMs. The only difference regards the fill-up technique: the additional binary feature is discarded, since it is already present in the corresponding filled-up TM. Hence, a filled-up RM is exactly the same as a backed-off RM.

2.4. Language model combination

Language models are built from the monolingual training data, as well as the target language of the parallel data. As the corpora available in the IWSLT evaluation come from a number of sources, we apply several methods to combine the LMs built on the available target language training corpora, rather than concatenating the data.

2.4.1. Mixture

Monolingual subcorpora can be combined into one mixture language model [11] by means of the IRSTLM toolkit [12]. The optimization of the internal mixture weights is achieved through a cross-validation approach on the same training data; hence no external development set is required. The mixture LM type can be loaded by Moses as any other LM type.

2.4.2. Linear interpolation

This technique, provided by the IRSTLM toolkit, consists in the linear interpolation of the n-gram probabilities from all component LMs. The optimal interpolation weights are computed by the EM algorithm which minimizes the perplexity on a given held-out development sample. The IRSTLM toolkit provides an interface that enables Moses to compute n-gram probabilities from interpolated LMs.

2.4.3. Log-linear interpolation

This technique, provided directly within the Moses toolkit, consists in the log-linear interpolation of the n-gram probabilities from all component LMs. The weight optimization is performed during the tuning of all Moses features.

3. English-French system

Our English-French systems are built upon a standard phrase-based system using the Moses toolkit [2], exploiting a huge amount of English-French bitexts and monolingual French training data. Each system features a statistical log-linear model including one phrase translation model [9] and one lexicalized reordering model, multiple French language models (LMs), as well as distortion, word, and phrase penalties.

The training data are composed from some of the corpora allowed by the IWSLT Evaluation Campaign organizers. As parallel data the following corpora were taken into account: Web Inventory of Transcribed and Translated Talks (version 2013-01) (TED) [13], 10^9-French-English (version 2) (Giga), English-French Europarl (version 7) (EP), Common Crawl (CC), MultiUN (UN), and the News Commentary (News) corpus as distributed by the organizers of the Workshop of Machine Translation (WMT). As monolingual data we use the entire monolingual news corpora (Full) distributed by WMT organizers for language model training. All texts were processed according to the language specific tokenization provided by Moses toolkit and kept case-sensitive. Statistics of the training corpora are reported in Table 1.

	unselected			selected		
Corpus	**Segm**	**En Words**	**Fr Words**	**Segm**	**En Words**	**Fr Words**
TED	155.5K	3.1M	3.2M	155.5K	3.1M	3.2M
Giga	22.5M	662.8M	774.7M	1.1M	23.8M	26.9M
UN	12.9M	361.6M	413.1M	257.7K	5.1M	5.6M
CC	3.2M	80.7M	88.0M	973.2K	23.2M	25.2M
EP	2.0M	55.6M	60.0M	240.9K	5.1M	4.8M
News	170.2K	4.4M	5.0M	51.1K	1.1M	1.3M
Full	84.0M	na	2.4T	na		

Table 1: Statistics of the parallel and monolingual data exploited for training our English-French systems. For the parallel data, statistics before and after data selection are reported. Symbols "T", "M" and "K" stand for 10^9, 10^6 and 10^3, respectively.

In order to focus the models toward a TED-specific domain and genre and to reduce the model size, data selection by means of the IRSTLM toolkit [12] is performed on the English-French bitexts, using the TED training data as in-domain data. Different amounts of data are selected from

each of the available out-of-domain corpora; statistics are reported in Table 1. A detailed description of the data selection procedure is provided in Section 2.1.

We construct five systems which exploit the training data in different ways to construct the component models. Details for these systems are provided in Section 3.1.

Most system parameters are kept fixed to allow a better comparison among the systems. Word alignments are computed by means of MGIZA++ on case-insensitive parallel texts to reduce data sparseness; casing information is re-introduced in order to estimate case-sensitive models, unless otherwise specified in the particular experiment. In all systems the maximum phrase length is set to 7 and the distortion limit is set to the default value of 6. We train 5-gram LMs with IRSTLM toolkit [12] in most cases; in other cases, KenLM [14] is used. Each language model is smoothed via the improved Kneser-Ney technique. Singleton n-grams of order three or higher are pruned.

The weights of the log-linear combination are optimized either via minimum error rate training (MERT) [15] or the Margin Infused Relaxed Algorithm (MIRA) [16, 17] on dev2010.

3.1. English-French submissions

As described in Section 3, we submit five systems which differ in the exploitation of the training data for the creation of TM, RM and LMs. We evaluate the performance of each system in Table 2 and use the results on tst2010 to select our primary submission. In our Primary, Contrastive 1, and Contrastive 2 systems, the dev2010 and tst2010 data are added to the TED training data after optimizing each system's feature weights, before evaluating their performances on the 2011, 2012, and 2013 test sets.

3.1.1. Primary

A backed-off TM is created combining a primary TM trained on TED training data (TED-TM) and a background TM trained on the selected training data (Slct-TM). The RM is constructed in a similar manner. A log-linear combination of two LMs is employed. The first LM is a mixture estimated from the in-domain TED training data (TED-LM) and the out-of-domain data-selected training data (Slct-LM). Additionally, a second Full-LM is estimated from the entire French monolingual corpora. Minimum Bayes Risk [18] (MBR) decoding technique, provided by Moses, is also exploited. Feature weights are averaged over three MERT optimizations.

3.1.2. Contrastive 1

This system replaces the backed-off TM of the primary system with a filled-up TM that exploits the same component TMs. Moreover, the MBR decoding technique is not applied. The feature weights are newly estimated averaging three distinct MERT optimizations.

3.1.3. Contrastive 2

This system aims at enhancing the primary system by further focusing its models to each specific talk that comprises the test set. Using the same optimized feature weights, we construct talk-specific translation, reordering, and language models and insert them with highest priority in their respective back-off and mixture models.

Given a talk to translate, we perform the data selection procedure described in Section 2.1, using the source text of the talk as seed data to extract the most similar portion from the data-selected parallel training data. Unlike the training phase, this selection is based on the English monolingual score only and a fixed amount of parallel data (about 3.5M English running words) were extracted.

Like the primary system, MBR decoding is applied. It is worth highlighting that this system is actually a collection of talk-specific instances working on their corresponding talk.

3.1.4. Enhanced Contrastive 2

In the post-evaluation activity, we performed an ad-hoc tuning of the system weights. For each talk of tst2010, we search for the optimal weights of the corresponding talk-specific system with our standard MERT procedure; then, all talk-specific weight sets are averaged and exploited for running the system over the official tst2011-2013. We also test this enhanced system on tst2010 in a fair manner: when translating a talk we exclude the corresponding set of optimal weight during the averaging action.

3.1.5. Contrastive 3

Following [10], the corpus specific TMs and RMs are combined according to the linear interpolation technique, but a different procedure is performed to find the mixing coefficients of the linear-interpolated TM and RM. A linear-interpolated LM is created by combining the corpus-specific LMs and its mixing coefficients are optimized by minimizing the perplexity on dev2010 target side using Expectation-Maximization by means of the IRSTLM toolkit. These interpolation weights are utilized as mixing coefficients for the linear-interpolated TM and RM. In this system we employ all LMs, estimated on the each of the 6 different domains, and the Full-LM combined in a log-linear fashion.

The system applies MBR decoding and case-insensitive models; therefore, a re-casing module estimated on the training data is attached to the translation system.

The whole set of the Moses features weights are optimized running the MIRA algorithm once.

3.1.6. Contrastive 4

This system differs from contrastive 3 only in the number of employed LMs; rather than using a log-linear combination of seven LMs, it utilizes only two: namely, TED-LM and Full-LM.

3.2. English-French results

Performance in terms of case-sensitive BLEU and TER of our primary (*P*) and contrastive (*C*) systems are reported in Table 2 and are compared to a simple TED baseline[2] (*B*). This baseline relies on TED training data only for the estimation of its TM, RM, and LM; the second Full LM is employed as well.

Figures referred to tst2010 were computed in-house, while those for tst2011-2013 are the official results provided by the organizers. As the official evaluation uses a slightly different text normalization procedure, the absolute scores are not directly comparable between different test sets; nevertheless, the relative difference among the systems are reliable.

In the result tables, the $^{\blacktriangledown}$ and $^{\triangledown}$ symbols beside the BLEU and TER scores indicate that the corresponding system performs significantly worse than the primary system with p-values not larger than 0.01 and 0.10, respectively. This annotation regards tst2010 only, for which the reference translations are available and hence the significance test can be performed.

	BLEU				TER			
	tst_{10}	**tst_{11}**	**tst_{12}**	**tst_{13}**	**tst_{10}**	**tst_{11}**	**tst_{12}**	**tst_{13}**
P	**34.11**	**38.41**	**39.51**	**37.69**	**0.472**	**0.420**	**0.406**	**0.441**
C_1	33.79$^{\triangledown}$	37.84	39.44	37.60	0.478$^{\blacktriangledown}$	0.426	0.409	0.441
C_2	31.90$^{\blacktriangledown}$	35.16	36.60	35.17	0.489$^{\blacktriangledown}$	0.443	0.429	0.458
C_3	34.03	28.99	29.69	26.36	0.479$^{\blacktriangledown}$	0.511	0.496	0.550
C_4	33.61$^{\triangledown}$	28.83	29.36	26.35	0.480$^{\blacktriangledown}$	0.511	0.498	0.548

Table 2: Results of the official English-French submissions evaluated on the IWSLT TED test sets. Symbols $^{\blacktriangledown}$ and $^{\triangledown}$ near to BLEU and TER scores on tst_{10} indicate that the system performs significantly worse than the primary system with p-values not larger than 0.01 and 0.10, respectively.

We can draw out some comments from the analysis of the official results. The primary system consistently outperforms the contrastive systems, and differences in scores are somehow kept constant. The improvement over the reference baseline system (shown in Table 3) is strongly significant, proving the effectiveness of the data selection approach applied

The low scores achieved by C_3 and C_4 on the 2011-2013 test sets are due to a misconfiguration of these systems when applied to the official data sets. After the official evaluation we translated the test sets with the corrected systems (C_3^* and C_4^*), and asked the organizers to re-evaluate them. New results are reported in Table 3. Scores for tst_{11}, tst_{12}, and tst_{13} have been computed by means of a different evaluation script; hence, figures in Tables 2 and 3 are not directly comparable.

On tst2010, all systems, but C_2, achieve very similar results in terms of both BLEU and TER. This is somehow expected, because the systems have very similar configurations. In terms of BLEU, a statistical test shows a slightly significant difference with respect to *P* only for C_1 and C_4, and only at p-value of 0.10. Instead, the differences in terms of TER are always significant.

Interestingly, from the results of C_3^* and C_4^*, we observe that the log linear combination of 6 language models does not improve the performance of the system, but instead it has negative effects on `tst2011` and `tst2013`. Use of out-domain language models diverge the "virtual domain" of interpolated TM and RM away from TED domain. The main difference between C_4^* and *P* is the way of combining TMs and RMs. *P* uses the back-off approach while C_4^* uses linear interpolation. This basically shows that back-off performs better than the linear interpolation technique for TED-talks data.

System C_2 is statistically worse than *P*. Our preliminary analysis showed that this system produced translation outputs about 4% shorter than *P*. Our feeling is that this is due to the exploitation of log-linear weights not specifically estimated for the talk-specific system. In order to confirm our conjecture, we translated the test sets with the enhanced system (C_2^*) described in Section 3.1, and its performance are reported in Table 3. It outperforms the primary system in terms of BLEU, but the differences are not significant, at least on tst2010. Instead, its performance in terms of TER are worse than those of the primary system; this is probably due to the fact that weight optimization aims at improving only the BLEU metric. A more balanced improvement could be achieved by tuning over several metrics.

	BLEU				TER			
	tst_{10}	**tst_{11}**	**tst_{12}**	**tst_{13}**	**tst_{10}**	**tst_{11}**	**tst_{12}**	**tst_{13}**
B	32.43$^{\blacktriangledown}$	35.77	36.95	34.56	0.489$^{\blacktriangledown}$	0.426	0.413	0.457
P	34.11	37.53	38.83	37.10	0.472	0.412	0.397	0.437
C_1	33.79$^{\triangledown}$	37.05	38.70	37.05	0.478$^{\blacktriangledown}$	0.418	0.401	0.433
C_2	31.90$^{\blacktriangledown}$	34.42	36.08	34.76	0.489$^{\blacktriangledown}$	0.436	0.421	0.450
C_2^*	34.28	38.72	39.80	37.68	0.486$^{\blacktriangledown}$	0.413	0.407	0.444
C_3^*	34.03	36.95	38.40	36.26	0.479$^{\blacktriangledown}$	0.423	0.405	0.449
C_4^*	33.61$^{\triangledown}$	37.28	38.14	36.42	0.480$^{\blacktriangledown}$	0.423	0.407	0.447

Table 3: Results of official and unofficial English-French submissions evaluated on the IWSLT TED test sets. C_2^*, C_3^*, and C_4^* are unofficial revised submissions. Scores for tst_{11}, tst_{12}, and tst_{13} have been computed by the organizers by means of an evaluation script partially different from the official one. Symbols $^{\blacktriangledown}$ and $^{\triangledown}$ near to BLEU and TER scores on tst_{10} indicate that the system performs significantly worse than the primary system with p-values not larger than 0.01 and 0.10, respectively.

4. English-Persian systems

The Persian-English (Fa[3]-En) and English-Persian (En-Fa) systems are built using similar configurations to our English-French system, described in Section 3. To relax the problem of token inconsistencies in Persian documents, we devel-

[2]System *B* was not submitted for the official evaluation, and therefore no results for tst2011-2013 are available.

[3]According to ISO 639-1 (*Codes for the representation of names of languages*), "Fa" is used as the abbreviation of Persian.

oped a Persian text normalizer that yields consistently better translation than the unnormalized text. Furthermore, to have a more precise Persian LM, we created a large Persian monolingual corpus by crawling feeds from several online news agencies. We show that the combination of specialized text normalization and a large LM trained on additional Persian data provides substantitive improvements over previous baselines.

4.1. Persian Text Normalization and Tokenization

Although there are some electronic standards for writing Persian, they are not uniformly followed by writers and software tools. These inconsistencies are observed in all existing textual resources, which cause many problems in natural language processing tasks. Several problems that commonly result in separate tokens for redundant types are described below.

Different character sets may be used for the same letter. Their appearance is virtually the same but different encodings exist for the characters. YEH(ي) and KAF (ك) are the best known cases in this category. On the other hand, some authors prefer to use imported letters from Arabic (e.g. أ) for writing the words borrowed from Arabic (رأى), while others use Persian letters (راى).

Diacritics are not typically written in the standard Persian text, but some authors decide to use them to reduce the ambiguty of the words. Although this makes the text more clear and understandable to the reader, not all authors use diacritic marks. Without proper preprocessing, the text processing system cannot classify different instances of the same word into one class.

Different word forms. This problem is mostly due to the word boundary ambiguation and different ways of putting space between different parts of words. For example, the word می روم (I am going) can be written in any of the following forms: می روم, میروم, and میروم. In the first and second forms, the prefix می and verb روم are separated using space and zero-width non-joiner (ZWNJ) characters, respectively; while in the last case, the prefix is attached to the verb.

To relax the problem of token inconsistency, we developed a Persian text normalizer and applied it on all of the Persian texts used in the experiments. This normalizer is published by the organizers and used to normalize all MT outputs and references before evaluating the systems in the English-Persian language pair tracks. A version of this tool was released for use with the IWSLT 2013 shared task. An enhanced version will be publicly available in the near future.

To measure the usefulness of the normalizer we develop two baseline systems using the normalized and non-normalized training data, and evaluate their translation quality in Table 4. The results show significant improvements in the final quality of the systems in both directions (1.5+ in BLEU scores and 2.7+ in TER). Furthurmore, comparing the vocabulary size of the normalized and non-normalized training corpus, shows more than 11 percent reduction in the number of unique words.

	BLEU		TER	
Metric	**Fa-En**	**En-Fa**	**Fa-En**	**En-Fa**
Baseline	12.47	9.13	0.734	0.758
Normalized	13.94	10.70	0.706	0.725

Table 4: Comparing the results of the normalized and unnormalized baselines on the IWSLT TED test set 2010.

4.2. Data Preparation

The data provided by the organizers for the Persian-English task is only the TED corpus; no additional parallel or monolingual corpora are provided for Persian. Though there are some other publicly available parallel corpora (namely, TEP [19], and PEN [20]), our initial experiments showed that using these corpora do not improve the baseline. Therefore, we decided not to use them in our submissions.

Regarding monolingual corpora, the Hamshahri corpus [21], used widely used in different Persian text processing tasks, has inconsistent sentence boundaries in such a way that in many cases one sentence is split into several lines, with no boundary markers in the corpus to capture the complete sentence.

Since this affects the language model creation and decreases the accuracy of the LM, we decided to create our own large Persian monolingual corpus with proper sentence boundaries. To create this corpus we extract texts from the archives of more than 20 online news agencies, mainly located in Iran. We extract the *body* of the news stories, as well as the *title*, *publish date*, and the *genre*, if available. The documents smaller than 1K are filtered out in this step. We normalize the documents using the aforementioned normalizer. The statistics of the corpus are presented in Table 5. This corpus will be publicly released at a future date.

		Tokens		Types	
Corpus	**Segm**	**English**	**Persian**	**English**	**Persian**
TED	77.1K	1.5M	1.7M	16.4K	20.8K
FBK	11.2M	na	309.2M	na	536.2K
FBK-slct	3.6M	na	50.1M	na	213K

Table 5: Statistics of the parallel and monolingual data exploited for training purpose in the English-Persian and Persian-English systems. Symbols "M" and "K" stand for 10^6 and 10^3, respectively. "FBK-slct" refers to the data selected portion of our internal Persian monolingual corpus.

For our Persian-English MT submission, we construct a common 5-gram mixture LM consisting of TED data, a subset of corpora from the LDC Gigaword fifth edition corpus, and the WMT News Commentary. From the Gigaword corpus, we select the articles from the Los Angeles Times/Washington Post, New York Times, and Washington Post/Bloomberg subcorpora. For the English-Persian task we used the TED training data (Persian side) and the monolin-

	BLEU				TER			
	$\mathbf{tst}_{10}$	$\mathbf{tst}_{11}$	$\mathbf{tst}_{12}$	$\mathbf{tst}_{13}$	$\mathbf{tst}_{10}$	$\mathbf{tst}_{11}$	$\mathbf{tst}_{12}$	$\mathbf{tst}_{13}$
B	12.47	16.39	12.80	12.49	0.734	0.678	0.88	0.876
P	14.62	18.85	14.40	14.32	0.703	0.664	0.861	0.858
C_1	–	–	–	14.47	–	–	–	0.858

Table 6: Results of submitted Persian-English runs evaluated on the IWSLT TED test sets.

	BLEU				TER			
	$\mathbf{tst}_{10}$	$\mathbf{tst}_{11}$	$\mathbf{tst}_{12}$	$\mathbf{tst}_{13}$	$\mathbf{tst}_{10}$	$\mathbf{tst}_{11}$	$\mathbf{tst}_{12}$	$\mathbf{tst}_{13}$
B	9.13	11.57	9.67	8.93	0.758	0.718	0.741	0.727
P	11.55	12.55	10.94	10.12	0.723	0.701	0.727	0.716
C_1	–	–	–	10.32	–	–	–	0.715

Table 7: Results of submitted English-Persian runs evaluated on the IWSLT TED test sets.

gual corpus described earlier.

4.3. English-Persian submissions

For both English-Persian and Persian-English tasks, we submitted a primary and a contrastive systems, which are briefly described in the following.

4.3.1. Primary

Our primary system uses the text normalization approach described in Section 4.1. For both the English-Persian and Persian-English submissions a TM is trained on TED training data, using similar configurations to our English-French systems, described in Section 3. For the Persian-English submission a log-linear combination of two LMs is employed. The primary LM is a 5-gram LM, trained on TED training data (English side), while the second LM is a 5-gram mixture LM consisting of TED data and the out-of-domain data-selected training data.

In the English-Persian direction the log-linear combination of LMs consist of three 5-gram LMs, trained on TED data, data selected from FBK Persian monolingual corpus, and whole FBK Persian monolingual corpus, respectively. As in our primary English-French submission, Minimum Bayes Risk decoding is exploited. Feature weights are optimized via Margin Infused Relaxed Algorithm (MIRA) on dev2010.

4.3.2. Contrastive

As mentioned earlier, the English-Persian language pair has few bitexts available for constructing a translation model. To measure the effects of adding additional in-domain corpora on translation quality, we augment the translation and reordering models with tst2011 and tst2012 and evaluate the results on tst2013 while retaining the log-linear weights of the original models.

4.4. English-Persian results

Our primary (*P*) and constrastive (*C*) results for Persian-English and English-Persian are reported in Tables 6 and 7, respectively. We compare the performance of our systems against a simple baseline (*B*), trained on the unnormalized TED data only. Scores on tst2010 clearly prove that our primary system highly outperforms the baseline.

The small amount of additional training data exploited in the contrastive system only gives a slight improvement in BLEU.

Long distance reorderings and the morphological richness of Persian are the two major problems in Persian-English SMT systems. On the other hand, hierarchical models are known to outperform the phrase-based systems for language pairs with differing word orders or long-distance reorderings. Our primary experiments in using hierarchical models for this langauge pair do not outperform the phrase-based baseline system, however. We will investigate this in more detail in future work.

One technique to overcome data sparsity due to morphological inflections is to perform unsupervised segmentation [22] and using the root forms for word alignment. However, in preliminary experiments we did not observe improvements over a baseline that only considers the surface form. One reason for this behavior may be due the fact that the suffixes they carry meaning that is lost during word alignment, which subsequently affects the quality of the extracted phrases. In the future we plan to try other morphological analysis strategies that better model the characteristics of Persian.

5. Conclusion

We presented the MT systems with which we participated in the IWSLT 2013 TED MT Evaluation Campaign. Our English-French systems benefited most from a "back-off" combination of in-domain and out-of-domain translation models, as well as a log-linear combination of two language model flavors: one which combines corpus-specific language models in a mixture model, and the other that concatenates all corpora and generates a gigantic LM.

Our English-Persian and Persian-English systems showed substantial improvements over a baseline provided by the workshop organizers, largely from improving the normalization and tokenization of Persian texts, as well as acquiring a large monolingual Persian news crawl corpus.

6. Acknowledgments

This work was partially supported by the EU-BRIDGE project (IST-287658), funded by the European Commission under the Seventh Framework Programme for Research and Technological Development.

7. References

[1] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, M. Federico, "Report on the 10th IWSLT Evaluation

Campaign," in *Proc. of the International Workshop on Spoken Language Translation*, December 2013.

[2] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic, 2007, pp. 177–180.

[3] C. Tillmann, "A Unigram Orientation Model for Statistical Machine Translation," in *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2004.

[4] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh system description for the 2005 IWSLT speech translation evaluation," in *Proc. of the International Workshop on Spoken Language Translation*, October 2005.

[5] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Morristown, NJ, USA: Association for Computational Linguistics, 2008, pp. 848–856.

[6] M. Cettolo, C. Servan, N. Bertoldi, M. Federico, L. Barrault, and H. Schwenk, "Issues in Incremental Adaptation of Statistical MT from Human Post-edits," in *Proceedings of the MT Summit XIV Workshop on Post-editing Technology and Practice (WPTP-2)*, Nice, France, September 2013.

[7] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *ACL (Short Papers)*, 2010, pp. 220–224.

[8] P. Nakov, "Improving English-Spanish Statistical Machine Translation: Experiments in Domain Adaptation, Sentence Paraphrasing, Tokenization, and Recasing," in *Workshop on Statistical Machine Translation, Association for Computational Linguistics*, 2008.

[9] A. Bisazza, N. Ruiz, and M. Federico, "Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation," in *International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, 2011, pp. 136–143.

[10] R. Sennrich, "Perplexity minimization for translation model domain adaptation in statistical machine translation," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Avignon, France: Association For Computational Linguistics, 2012, pp. 539–549.

[11] M. Federico and R. De Mori, "Language modelling," in *Spoken Dialogues with Computers*, R. D. Mori, Ed. London, UK: Academy Press, 1998, ch. 7, pp. 199–230.

[12] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models," in *Proceedings of Interspeech*, Brisbane, Australia, 2008, pp. 1618–1621.

[13] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks," in *Proceedings of the Annual Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012.

[14] K. Heafield, "KenLM: Faster and Smaller Language Model Queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, United Kingdom, July 2011, pp. 187–197.

[15] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, E. Hinrichs and D. Roth, Eds., 2003, pp. 160–167.

[16] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[17] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki, "Online large-margin training for statistical machine translation," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007, pp. 764–773.

[18] S. Kumar and W. Byrne, "Minimum bayes-risk decoding for statistical machine translation," in *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2004.

[19] M. Pilevar, H. Faili, and A. Pilevar, "Tep: Tehran english-persian parallel corpus," in *Computational Linguistics and Intelligent Text Processing*, ser. Lecture Notes in Computer Science, A. Gelbukh, Ed. Springer Berlin Heidelberg, 2011, vol. 6609, pp. 68–79.

[20] M. A. Farajian, "Pen: Parallel english-persian news corpus," in *Proceedings of 2011 International Conference on Artificial Intelligence (ICAI'11)*, Las Vegas, NV, 2011.

[21] A. AleAhmad, H. Amiri, E. Darrudi, M. Rahgozar, and F. Oroumchian, "Hamshahri: A standard persian text collection," *Know.-Based Syst.*, vol. 22, no. 5, pp. 382–387, July 2009.

[22] M. Creutz and K. Lagus, "Inducing the morphological lexicon of a natural language from unannotated text," in *International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, 2005.

The Heidelberg University Machine Translation Systems for IWSLT2013

Patrick Simianer, Laura Jehl, Stefan Riezler

Department of Computational Linguistics
Heidelberg University, Germany
{simianer,jehl,riezler}@cl.uni-heidelberg.de

Abstract

We present our systems for the machine translation evaluation campaign of the *International Workshop on Spoken Language Translation* (IWSLT) 2013. We submitted systems for three language directions: German-to-English, Russian-to-English and English-to-Russian. The focus of our approaches lies on effective usage of the in-domain parallel training data. Therefore, we use the training data to tune parameter weights for millions of sparse lexicalized features using efficient parallelized stochastic learning techniques. For German-to-English we incorporate syntax features. We combine all of our systems with large language models. For the systems involving Russian we also incorporate more data into building of the translation models.

1. Introduction

This paper describes Heidelberg University (HDU)'s machine translation (MT) systems built for the IWSLT 2013 MT evaluation campaign. We submitted results for three translation directions: German-to-English, Russian-to-English and English-to-Russian.

Our German-to-English system does not use any parallel data other than the data provided by the organizers. Hence, we try to use this small amount (as compared to data available for other domains) of parallel data as effectively as possible by using the full training data to tune models with millions of features, e.g. lexicalized features derived from translation rules. The model parameters are learned by a perceptron algorithm in a pairwise-ranking framework with sharding for parallelization. See subsection 1.2 for a full explanation of this learning framework and a brief description of the features. For German-to-English we additionally experimented with the soft-syntactic constraints of [1] to determine whether or not they can improve spoken language translation.

The systems for the Russian-to-English and English-to-Russian directions were built using the same techniques, but with additional parallel training data for the translation model estimation, as the baseline systems are of low quality – with BLEU scores far below the 20% mark.

All systems make use of large language models (LM) at test-time. We do not use any data filtering or domain adaption techniques for any of our systems.

1.1. Technical System Commonalities

The systems described in this paper are all based on the hierarchical phrase-based paradigm for statistical machine translation [2] using the `cdec`[1] [3] decoding framework.

Pre- and post-processing, i.e. (de-)tokenization and recasing were done using the moses toolkit[2]. The recaser was always trained with default parameters using solely the target side of the provided parallel data (parallel transcriptions of TED talks) – even if the rest of the system was trained using more data.

Word alignments for the parallel data were built according to a variant of IBM's model 2 as described in [4] using the associated implementation[3]. To obtain many-to-many alignments, models for both directions were built and the resulting alignments were symmetrized using the *grow-diag-final-and* heuristic. We applied a Dirichlet prior on the lexical translation distributions and favored alignments that are close to the monotonic diagonal using default parameters for all language pairs.

Hiero-style grammars – allowing only a single type of non-terminal `X` – were built using the suffix array technique described in [5] with parameters as in [2].

All Language models use modified Kneser-Ney smoothing and are estimated using the implementation[4] of [6].

System-selection was carried out using either a tournament-like subjective evaluation of several annotators on a random sample of 30 translations for each round; or simply based on automatic scoring results on the development test set, which was `tst2010` for all language pairs.

Evaluation scores reported in this paper are calculated with cased and tokenized text using `MultEval`[5], so that our results are comparable to the official results of the evaluation campaign of IWSLT 2012. All MERT results we report are averaged scores over three runs, to overcome optimizer instability (see [7]). All other methods discussed in this paper are stable in this respect.

[1] `http://www.cdec-decoder.org/`
[2] `https://github.com/moses-smt/mosesdecoder`
[3] `https://github.com/clab/fast_align`
[4] `http://kheafield.com/code/kenlm/estimation/`
[5] `https://github.com/jhclark/multeval`

Get data for Z shards, each including S sentences;
distribute to machines.
Initialize $\mathbf{v} \leftarrow \mathbf{0}$.
for epochs $t \leftarrow 0 \dots T-1$: **do**
  for all shards $z \in \{1 \dots Z\}$: **parallel do**
    $\mathbf{w}_{z,t,0,0} \leftarrow \mathbf{v}$
    for all sentences $i \in \{0 \dots S-1\}$: **do**
      Decode i^{th} input with $\mathbf{w}_{z,t,i,0}$.
      for all pairs $j \in \{0 \dots P-1\}$: **do**
        $\mathbf{w}_{z,t,i,j+1} \leftarrow \mathbf{w}_{z,t,i,j} - \eta \nabla l_j(\mathbf{w}_{z,t,i,j})$
      end for
      $\mathbf{w}_{z,t,i+1,0} \leftarrow \mathbf{w}_{z,t,i,P}$
    end for
  end for
  Stack weights $\mathbf{W} \leftarrow [\mathbf{w}_{1,t,S,0}| \dots |\mathbf{w}_{Z,t,S,0}]^T$
  Select top K feature columns of $\mathbf{W}$ by ℓ_2 norm
  for $k \leftarrow 1 \dots K$ **do**
    $\mathbf{v}[k] = \frac{1}{Z} \sum_{z=1}^{Z} \mathbf{W}[z][k]$.
  end for
end for
return $\mathbf{v}$

Figure 1: Pairwise ranking-optimization algorithm with ℓ_1/ℓ_2 regularization that enables the use of large tuning sets and millions of sparse features. The data is divided into evenly sized shards, which can then be processed in parallel. The core of the algorithm is the stochastic gradient update in the innermost loop. After all shards are finished, the regularization selects the top K features by ℓ_2 norm of weights over shards for another epoch.

1.2. Tuning on the Training Set

To effectively make use of the limited in-domain parallel training data we employ the technique of [8] to train models with a large number of features using the full training set. The parameters of the translation and language models as well as other dense features are trained simultaneously.

While the amount of in-domain parallel data provided is small compared to other data sets, tuning on this amount of data is a non-trivial task, as most approaches are tailored to use a few thousand parallel segments.

The approach described in [8] enables the use of millions of sparse features using hundreds of thousands parallel segments. The algorithm is shown in Figure 1. The core of this algorithm is the stochastic gradient update in the innermost loop. With this, the algorithm seeks to minimize the following loss in a pairwise-ranking setup (see e.g. [9]):

$$l_j(\mathbf{w}) = (-\langle \mathbf{w}, \bar{\mathbf{x}}_j \rangle)_+,$$

where $\bar{\mathbf{x}} = \mathbf{x}^{(1)} - \mathbf{x}^{(2)}$; $\mathbf{x}$ are feature representations of translations; $\mathbf{x}^{(1)}$ is preferred over $\mathbf{x}^{(2)}$ by a local approximation of the BLEU score as discussed in detail in [10][6]. Taking the derivative of this loss function leads to a standard perceptron update.

As [11] show, the theory behind the perceptron algorithm still holds – as an instance of stochastic gradient descent – when training data is sharded and resulting parameters are averaged. [8] extend this by adopting ℓ_1/ℓ_2 regularization, which limits the number of features in the model and thus improves efficiency. For use with a single set of parallel segments (e.g. a standard development set) the whole algorithm reduces to the innermost loop. In this case, the weight vectors of all epochs are averaged to obtain the final model, see [12] for a theoretical and empirical background.

Several sparse feature templates are used, all of which are derived from translation rules:

- rule id: Each rule is a feature in the new model.
- rule n-grams: n-grams of source and target side of rules (including non-terminals); we use bigrams for both source and target.
- rule shape: Each rule is represented by its shape defined by its composition of terminal and non-terminals, see [8] for an example.

We call this method "dtrain", no matter what amount of training data is used for tuning. Please note that in this paper dtrain is always combined with the sparse feature set as listed above.

To prevent overfitting on the training set, we employ the "folding" method described in [13] when building translation and language models for shards. For each shard, separate language and translation models are built from all available data, but excluding the data of the current shard.

2. German-to-English

For German-to-English we only use the provided parallel TED data for estimation of the translation model: 138,499 parallel segments, with 2,639,101 German and 2,762,380 English tokens after pre-processing. German compound words were split using the empirical approach described in [14]. The compound splitting model was trained on the German side of the parallel corpus using the defaults of the implementation in the moses toolkit.

As English is the prevalent language in machine translation evaluation campaigns, there is a wide range of freely available English corpora to build large language models. We used the data listed in Table 2 to build a 5-gram language model, which was only used for evaluation at test time. Another 5-gram LM was built from the *LDC2011T07* corpus (English Gigaword Fifth Edition, "Giga") alone. For tuning and development we used a 4-gram language model built from the provided monolingual TED data.

2.1. Syntax Features

In decoding with the hierarchical phrase-based approach there is the possibility to reward proper use of syntax on source- or target-side, as hierarchical derivations are built for both sides during the process. [1] introduce soft-syntactic constraints to reward partial derivations which correspond

[6] Our variant is *grounded* and *BP-smoothed*, as we found superior performance compared to other variants.

System	TED 4-gram LM	Giga 5-gram LM	Large 5-gram LM
baseline	26.7	-	-
mert-dev	26.7	28.1	28.4
dtrain-dev	27.6	28.8	28.8
dtrain-train(clustered)*	28.0	29.4	**29.6**
dtrain-train+soft-syntax†	28.1	28.9	-
dtrain-train+	28.1	29.2	**29.6**

Table 1: German-to-English evaluation results on `tst2010` in % BLEU-4. MERT was used to tune the dense weights of the hierarchical phrase-based system using the `dev2010` set. dtrain exploits the full sparse feature set for `dev2010`. Systems below the double dash are large-scale experiments utilizing the full training set for tuning. We submitted three systems: * primary, † contrastive #1, + contrastive #2. Our best results are marked in bold.

Corpus	Segments	Tokens
10^9 FR-EN Release2	22,520,400	575,667,242
Europarl v7 (merged)	2,342,410	58,567,624
News Comm. v8 (merged)	272,508	6,363,229
News Crawl 2007	3,782,548	77,701,721
News Crawl 2008	12,954,477	265,801,031
News Crawl 2009	14,680,024	300,118,377
News Crawl 2010	6,797,225	136,709,612
News Crawl 2011	15,437,674	309,687,553
News Crawl 2012	14,869,673	299,023,941
UN corpus	14,118,662	343,386,910
LDC2011T07	187,848,540	4,872,200,262
$\sum$	295,624,141	7,245,227,502

Table 2: Counts of corpora used for the large English language model. English sides of parallel data sets and corresponding monolingual data were merged by repeating each unique segment the maximum number of times it has occurred in any of the files involved in the process.

to syntactic constituents on the source side. This is done through features which indicate proper syntactic structures in the parse of the source sentence. This way, the system can learn whether or not it is beneficial to the evaluation metric optimized in tuning to match or cross[7] syntactic constituents (e.g. NP, VP etc.). For each rule application, the feature searches a pre-computed syntax tree for a constituent matching its span. We used the *Stanford Parser*[8] for pre-computing the German parses. This approach is considered "soft", as it is feature-based and therefore only encodes preferences, not enforcing hard constraints.

2.2. Experiments

We conducted several preliminary experiments with this language pair, the results were carried over to our other systems: A search for a good trade-off between speed and performance for the shard size (we found 2,200 segments per shard to be a good value) and a coarse grid search for the optimal learning rate of the pairwise-ranking optimization (dtrain). Our main results for German-to-English are shown in Table 1. "mert-dev" is a simple recreation of the official baseline using our hierarchical phrase-based system, including our pre- and post-processing. "dtrain-dev" uses our method for pairwise-ranking optimization on the same development set (`dev2010`) with the full sparse feature set, i.e. rule id, rule bigrams and rule shape features. We see that this already gives an improvement of about 1.0 BLEU% point over mert-dev. Adding the large language model when evaluating leads to further improvements.

For each of the experiments conducted on the training set ("dtrain-train*") the full sparse feature set was used. "dtrain-train(clustered)" is a system where we clustered the talks in the training set according to their assigned keywords, following the intuition of [15] that data should be divided by natural "tasks" for optimal learning. We chose the number of clusters such that the shard size was comparable to the optimal shard size found in preliminary experiments. This resulted in a use of about 70% of the original training data, as some clusters were just too small to be included. The second system ("dtrain-train+soft-syntax") utilized the training set, partitioned into equally sized shards (2,200 segments per shard), including the soft-syntactic constraint features as described in subsection 2.1 in addition to the sparse features. We used all available 20 non-terminal symbols, resulting in 40 features overall. Our third submitted system for German-to-English, "dtrain-train", is equivalent to the previous described system, but does not make use of the soft-syntactic constraints. We find very similar performance in all of our training set experiments, with the exception that the system with syntax features is falling behind when scaling to larger language models (we did not use the largest language model with this system due to time constraints).

Using the large language model and our best system we see an improvement of 2.9 BLEU% points over the official baseline.

[7] Two features are defined for each non-terminal label.

[8] `http://nlp.stanford.edu/software/lex-parser.shtml`

Corpus	Segments	RU Tokens	EN Tokens
Common Crawl	878,386	17,399,366	18,772,065
Yandex 1M corpus	1,000,000	20,237,417	22,796,278
News Commentary v8	150,217	3,269,668	3,488,752
Wiki Headlines	444,532	917,277	1,045,416
TED parallel data	128,592	2,218,547	2,575,289
$\sum$	2,601,727	44,042,275	48,677,800

Table 3: Corpora that were combined for the extended Russian-to-English translation model.

3. Russian-to-English

[16] show that translating into or from Russian is harder than translation of other Romanic or Germanic languages, at least in the TED domain. The provided parallel and monolingual TED training data is of similar size as for the German-English language pair. Therefore, we used additional data besides the official parallel TED data for building the translation model. The data sets used for this are listed in Table 3. We reused the English language models from the German-to-English systems.

3.1. Experiments

The cascade of experiments conducted for the Russian-to-English direction is shown in Table 4. We approximately match the baseline using our standard hierarchical phrase-based system (mert-dev). There are small improvements using the sparse feature set and utilizing the pairwise-ranking optimization (dtrain-dev). When enabling the large language model while tuning, we achieve additional 0.3 BLEU% points improvement. We see big gains with the enlarged translation model, at least 2.0 points for all systems.

Increasing the amount of training data for the pairwise-ranking optimization does not improve over the best system on the small development set when using the small translation model.

The best result, with an improvement of 3.7 BLEU% points over our baseline, was achieved by scaling up all aspects of the machine translation system, the language and translation models, as well as the training data size for dtrain. But note that this system only used 42,000 segments of the available TED training data, as the "folding" technique described in subsection 1.2 is very time consuming when used in combination with larger amounts of parallel data.

4. English-to-Russian

English-to-Russian is a very challenging translation direction in the TED domain, which is reflected by low baseline evaluation scores – the baseline reported in [17] is about 12.5 BLEU% points. Hence, we chose to use more parallel training data for the English-to-Russian system, the same data as used for the Russian-to-English system. We built a 4-gram language model from the provided monolingual data and a large Russian 5-gram language model from the data listed in Table 5.

Corpus	Segments	Tokens
Common Crawl	878,386	17,399,366
News Comm. v8 (Russian tgt)	150,217	3,269,668
News Comm. v8 (Russian)	183,083	3,649,222
Yandex 1M Corpus	1,000,000	20,237,417
News Crawl 2008	38,195	587,775
News Crawl 2009	91,119	1,331,658
News Crawl 2010	47,818	652,288
News Crawl 2011	9,945,918	142,629,530
News Crawl 2012	9,789,861	143,407,485
TED Russian data	136,101	1,859,376
$\sum$	22,260,698	335,023,785

Table 5: Data for the large Russian language model.

4.1. Experiments

Results for the English-Russian experiments are given in Table 6. Our MERT-trained baseline with dense features ("mert-dev") achieves about the same performance as the official phrase-based baseline. Using only the dense feature set, this system does not benefit strongly from using the enlarged translation model. We manage to improve over MERT using sparse features and the pairwise-ranking optimization on the development set ("dtrain-dev"). If the large Russian language model is used during tuning and evaluation, we obtain another improvement of 0.2 points. Our best results are obtained using dtrain on the development set with sparse features and the extended translation model. While the improvement using the small 4-gram language model is not large at 0.3 points, the combination of the large translation model and the large language model for evaluation is very significant and leads to an overall improvement of 2.4 BLEU% points over our baseline.

Using the full training data for dtrain leads to inferior results for this translation direction. The reasons for this remain to be investigated. Therefore, we did not try to use the enlarged translation model with this approach.

5. Conclusions

For all language pairs we considered, our baseline hierarchical phrase-based systems perform on a par with the official baselines that build upon the phrase-based moses toolkit. Adding sparse features derived from translation rules helps for all language pairs, even if their parameters are estimated on a small development set. Scaling up in terms of training data for the pairwise-ranking optimization leads to further improvements, with the notable exception of our English-to-Russian system, where we have a weak translation model. Increasing the size of the language model is a trivial but effective improvement, even more so without applying any filtering or domain adaptation techniques. A drawback to these

System	TED 4-gram LM	Large 5-gram LM
baseline	17.2	-
mert-dev	17.0	17.5
dtrain-dev	17.2	17.8
dtrain+large LM^{+}	-	18.1
dtrain+large TM	*19.2*	*19.8*
dtrain+large TM+large LM	-	*20.1*
dtrain-train†	17.7	18.4
dtrain-train+large LM+large TM*	-	**20.7**

Table 4: Results for Russian-to-English systems on `tst2010`. We submitted three systems: * primary, † contrastive #1, $^{+}$ contrastive #2. Our best result is marked in bold. Systems in italics were not available for the submission deadline.

System	TED 4-gram LM	Large 5-gram LM
baseline	12.5	-
mert-dev	12.4	13.1
mert-dev+large TM	12.5	13.5
dtrain-dev	12.8	13.7
dtrain-dev+large LM	-	13.9
dtrain-dev+large TM*	13.1	**14.8**
dtrain-dev+large LM+large TM	-	14.6
dtrain-train†	11.8	13.2

Table 6: Results for English-Russian systems on `tst2010` in % BLEU-4. * denotes the primary system for this language pair; † the contrastive system. Our best result is marked in bold.

simple improvements is the strongly increased computational requirements, although most of the tools we used scale up nicely.

6. Official Results

Table 7 shows the official results for our submitted systems for the three translation directions we participated in. All systems use the largest language model built for their respective target language. Unlike the development and training sets, the source for `tst2013` contained disfluencies, thus the organizers calculated BLEU scores using two different reference sets, one with and one without disfluencies. Our systems seem robust, as both of the scores are nearly identical, e.g. our primary system for German-to-English scores 23.06 without disfluencies and 22.91 with disfluencies in the reference. Our primary submission for Russian-to-English was erroneous, using a small scale translation model when the large TM was used for tuning. Corrected, the primary Russian-to-English system shows good performance, scaling up in all aspects of the translation system: language model (used for tuning and evaluation), translation model, feature set and tuning data size. The English-to-Russian system depicts the same gap between small and large tuning set size as shown on the development test set.

7. Acknowledgements

The research presented in this paper was supported in part by DFG grant "Cross-language Learning-to-Rank for Patent Retrieval".

8. References

[1] Y. Marton and P. Resnik, "Soft syntactic constraints for hierarchical phrase-based translation," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, Columbus, OH, 2008.

[2] D. Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, 2007.

[3] C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, P. Blunsom, H. Setiawan, V. Eidelman, and P. Resnik, "cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models," in *Proceedings of the ACL 2010 System Demonstrations*, Uppsala, Sweden, 2010.

[4] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Com-*

German-to-English	tst2011	tst2012	tst2013
primary (dtrain-train(clustered))	-	-	23.06 (24.07)
contrastive #1 (dtrain-train+soft-syntax)	-	-	22.36 (23.38)
contrastive #2 (dtrain-train)	-	-	22.94 (23.93)
Russian-to-English	tst2011	tst2012	tst2013
primary (dtrain-train+large LM+large TM)	20.16 (21.30)	18.21 (19.40)	20.58 (21.50)
primary (corrected)	22.91 (24.45)	20.16 (21.53)	23.78 (25.00)
contrastive #1 (dtrain-train)	20.00 (21.19)	18.20 (19.37)	20.56 (21.50)
contrastive #2 (dtrain-dev+large LM)	19.87 (20.99)	18.08 (19.18)	20.41 (21.45)
English-to-Russian	tst2011	tst2012	tst2013
primary (dtrain-dev+large TM)	15.53 (15.61)	13.76 (13.83)	15.87 (15.95)
contrastive #1 (dtrain-train)	14.20 (14.24)	12.83 (12.87)	14.56 (14.62)

Table 7: Official results. Scores were calculated using `mteval-v13a` on cased (lowercased in parenthesis) and detokenized text.

putational Linguistics: Human Language Technologies (NAACL-HLT'13), Atlanta, GA, 2013.

[5] A. Lopez, "Hierarchical phrase-based translation with suffix arrays," in *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, Prague, Czech Republic, 2007.

[6] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, "Scalable modified Kneser-Ney language model estimation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL'13)*, Sofia, Bulgaria, 2013.

[7] J. Clark, C. Dyer, A. Lavie, and N. Smith, "Better hypothesis testing for statistical machine translation: Controlling for optimizer instability," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL'11)*, Portland, OR, 2011.

[8] P. Simianer, S. Riezler, and C. Dyer, "Joint Feature Selection in Distributed Stochastic Learning for Large-Scale Discriminative Training in SMT," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12)*, Jeju Island, Korea, 2012.

[9] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*, Edinburgh, Scotland, 2011.

[10] P. Nakov, F. Guzman, and S. Vogel, "Optimizing for sentence-level bleu+1 yields short translations," in *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, Bombay, India, 2012.

[11] M. A. Zinkevich, M. Weimer, A. Smola, and L. Li, "Parallelized stochastic gradient descent," in *Proceedings of the 24th Annual Conference on Neural Information Processing Sytems (NIPS'10)*, Vancouver, Canada, 2010.

[12] M. Collins, "Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms," in *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP'02)*, Philadelphia, PA, 2002.

[13] J. Flanigan, C. Dyer, and J. Carbonell, "Large-scale discriminative training for statistical machine translation using held-out line search," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'13)*, Atlanta, GA, 2013.

[14] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proceedings of the 10th Conference on European chapter of the Association for Computational Linguistics (EACL'03)*, Budapest, Hungary, 2003.

[15] P. Simianer and S. Riezler, "Multi-task learning for improved discriminative training in SMT," in *Proceedings of the ACL 2013 Eighth Workshop on Statistical Machine Translation (WMT'13)*, Sofia, Bulgaria, 2013.

[16] G. Neubig, K. Duh, M. Ogushi, T. Kano, T. Kiso, S. Sakti, T. Toda, and S. Nakamura, "The NAIST machine translation system for IWSLT 2012," in *International Workshop on Spoken Language Translation (IWSLT'12)*, Hong Kong, 2012.

[17] M. Cettolo, C. Girardi, and M. Federico, "Wit3: Web inventory of transcribed and translated talks," in *Proceedings of the 16thConference of the European Association for Machine Translation (EAMT'12)*, Trento, Italy, 2012.

The UEDIN English ASR System for the IWSLT 2013 Evaluation

Peter Bell, Fergus McInnes, Siva Reddy Gangireddy,
Mark Sinclair, Alexandra Birch, Steve Renals

School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK
{peter.bell,fergus.mcinnes,a.birch,s.renals}@ed.ac.uk,
{s.gangireddy,m.sinclair-7}@sms.ed.ac.uk

Abstract

This paper describes the University of Edinburgh (UEDIN) English ASR system for the IWSLT 2013 Evaluation. Notable features of the system include deep neural network acoustic models in both tandem and hybrid configuration, cross-domain adaptation with multi-level adaptive networks, and the use of a recurrent neural network language model. Improvements to our system since the 2012 evaluation – which include the use of a significantly improved n-gram language model – result in a 19% relative WER reduction on the `tst2012` set.

1. Introduction

We report on experiments carried out for the development of automatic speech recognition (ASR) systems on the English datasets of the International Workshop on Spoken Language Translation (IWSLT) 2013. We report our work on the new TED German task in an accompanying paper [1] since the development of the two systems was largely independent. Work on our machine translation system may be found in [2]. Significant changes to the English ASR system since 2012 include improvements to our baseline language models, described in Section 2.1, and the use of recurrent neural network language models, described in Section 2.2. The acoustic models are described in Section 3 – the main addition is that we now use deep neural networks in a hybrid configuration, and apply automatic voice activity detection to the `tst2013` test set.

2. Language modelling

The ASR system used Kneser-Ney smoothed N-gram language models for decoding and lattice rescoring, and a recurrent neural network (RNN) language model for a final rescoring stage based on N-best lists. These models are described in the subsections below.

This work was supported by the European Union under the FP7 projects inEvent (grant agreement 287872) and EU-Bridge (grant agreement 287658), and by EPSRC Programme Grant grant EP/I031022/1, *Natural Speech Technology*.

2.1. N-gram models

The N-gram language models were obtained by interpolating individual modified Kneser-Ney discounted LMs trained on the small in-domain corpus of TED transcripts and the larger out-of-domain (OOD) sources. The OOD sources were Europarl (v7), News Commentary (v7), News Crawl (2007 to 2011) and Gigaword (Fifth Edition).

The News Crawl and Gigaword sources in particular contained a wide variety of phenomena such as money amounts and other numerical expressions, abbreviations, and listed and tabulated information, which required normalisation to create data resembling spoken word sequences. Considerable effort was put into developing appropriate text normalisation scripts. Starting from the scripts used in LM training for the IWSLT 2012 evaluation, over 1000 lines of Perl code and 1400 abbreviation entries were added (expanding the original files by more than 50%). The processing applied to the data can be summarised as follows.

1. Remove documents that are not of type *story*, strip out markup and split text into sentences (required for Gigaword only).
2. Eliminate duplicate lines (common in some newswire sources, where multiple copies or variants of the same story may occur).
3. Convert Unicode characters and encodings for fractions, symbols etc into standard ASCII forms such as "1/4" (for subsequent conversion to words).
4. Filter out newswire datelines, e.g. "LONDON, Nov 2", and other extraneous material.
5. Normalise punctuation, abbreviations, units of measurement etc.
6. Convert numerical expressions to words.
7. Remove punctuation and convert to lower-case without diacritics.
8. Convert British to American English spellings and correct some common spelling errors.

The vocabulary for the ASR system was defined so as to include all words occurring in the in-domain training corpus (other than words which occurred only once and were not in a standard dictionary) and all words exceeding specified occurrence count thresholds in the OOD corpora, while remaining below the maximum of 64K words imposed by the version of HDecode in use here. The vocabulary size was 62,522.

Initialisms included in the vocabulary were treated as single words for LM purposes, e.g. "u.s." (with the dots retained to distinguish them from words such as "us"). Once the vocabulary had been defined, out-of-vocabulary initialisms were broken into single letters, e.g. "m. f. n.", so as to be modelled as sequences of in-vocabulary words (letter names) rather than treated as OOV.

In view of the mismatch in content and style between the target domain (TED talks) and the OOD data, a data selection process [3, 4] was applied to the OOD corpora to obtain an appropriate subset of data for LM training. The set of out-of-domain data D_S was chosen by computing a cross-entropy difference (CED) score for each sentence s:

$$D_S = \{s | H_I(s) - H_O(s) < \tau\} \quad (1)$$

where $H_I(s)$ is a cross-entropy of a sentence with a LM trained on in-domain data, $H_O(s)$ is a cross-entropy of a sentence with a LM trained on a random subset of the OOD data of similar size to the TED corpus, and τ is a threshold to control the size of D_S

Language models were trained on the in-domain and OOD data using the SRILM toolkit [5], and were interpolated with weights optimised on the TED development set (dev2010 and tst2010: total 44,456 words).

Perplexities on the development set with 3-gram and 4-gram models trained on the TED corpus and selected OOD data are shown in Table 1. Selecting 25% of the OOD sentences yielded an OOD training set of 751M words; setting the CED threshold to 0 gave a smaller but more targeted set of 312M words, which gave a lower perplexity on the TED data than the 751M word set when used alone to train the LM, but a slightly higher perplexity after interpolation with the TED LM. The perplexities obtained here are substantially lower than the values of 160 (3-gram) and 159 (4-gram) with the LMs used in our IWSLT 2012 system [6], which were trained using a much smaller set of OOD data with no CED filtering.

The LMs finally used in the ASR system were the TED+312MW trigram model (for decoding) and the TED+312MW 4-gram model (for lattice rescoring). The amounts of data from the respective sources used in these LMs are shown in the "Selected" column of Table 2. Comparison with the total sizes of the source corpora (after text normalisation) given in the preceding column shows that the proportion of data selected by the CED criterion ranged from 8% for the Gigaword corpus to 15% for News Commentary.

Language model	Perplexity
TED 3-gram	183.2
OOD (312MW / 751MW) 3-gram	133.5 / 138.3
TED+OOD (312MW / 751MW) 3-gram	125.1 / 124.9
TED 4-gram	179.9
OOD (312MW / 751MW) 4-gram	123.9 / 126.4
TED+OOD (312MW / 751MW) 4-gram	114.9 / 113.4

Table 1: Perplexities of N-gram language models on TED development set.

Corpus	Total	Selected
TED	2.4M	2.4M
Europarl	53.1M	6.3M
News Commentary	4.4M	0.7M
News Crawl	693.5M	72.9M
Gigaword	2915.6M	232.9M
OOD total	3666.6M	312.8M

Table 2: Numbers of words in LM training sets.

2.2. RNN models

Neural network language models have shown to consistently improve the word error rates (WER) of LVSCR tasks [7, 8, 9]. For this year's evaluation, we investigated the effectiveness of RNN LMs for TED lecture transcription. To study the effectiveness of RNNs we rescored the n-best hypothesis using RNNs trained on in-domain and different subsets of out-of-domain (OOD) data, shown in Table 3, selecting according to the CED score as in Section 2.1 In-domain data consists of 2.4M tokens. Since it is very difficult to train the RNNs on large amounts of OOD data, we restrict the maximum size of OOD data to 30M.

The number of hidden neurons ranged from 300 to 500 and number of classes in the output layer was 300. Models are trained using RNN training tool of [10]. Table 4 shows the perplexity (PPL) and WER on on development data provided by IWSLT evaluation campaign. We can observe that rescoring the n-best hypothesis with the RNNs reduce the WER by 0.8%. We choose the best model from this experiments to rescore the n-best hypothesis from `tst2011`, `tst2012` and the `tst2013` test sets. The interpolation weight between n-gram and RNNLM is optimised on devel-

Table 3: Subsets of OOD data

#Words	#Sentences	Threshold(τ)
5M	664.2K	-1.14
10M	1156.7K	-0.963
15M	1596.7K	-0.862
20M	2011.3K	-0.79
25M	2412.6K	-0.733
30M	2792.4K	-0.687

Table 4: Perplexity and WER on development data

Tokens	Vocabulary	PPL	WER(%)
n-gram	-	-	15.6
7.4M	47.7K	171.56	15.2
12.4M	54.8K	161.66	15.2
17.4M	61.7K	147.17	15.0
22.4M	68K	142.22	14.9
27.4M	74.3K	133.5	14.8
32.4M	80K	**126.0**	**14.8**

opment data, to minimise WER.

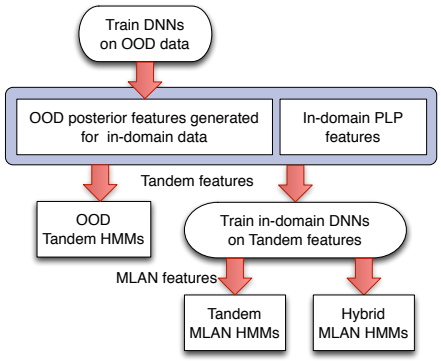


Figure 1: Tandem and hybrid MLAN training

3. Acoustic modelling

For the acoustic modelling components of the system, we used a setup identical to that described in [11], where more details may be found. Briefly, we used a combination of tandem and hybrid deep neural network (DNN) systems trained on a corpus of in-domain TED talks, incorporating out-of-domain data of multi-party meetings from the AMI corpus using the multi-level adaptive networks (MLAN) technique [12]. Compared to our 2012 system, the main addition is the use of DNNs with MLAN features in the hybrid framework. We describe this further below. Additionally, unlike earlier test sets from the IWSLT evaluation, the 2013 test set was not provided with a manually derived segmentation; we therefore employed an automatic segmentation system, described in Section 3.3.

3.1. Training data

For in-domain training data, we used 813 TED talks recorded prior to the end of 2010. The talks were segmented and aligned to the crowd-sourced transcriptions available online using a lightly-supervised technique described in [13]. This produced 143 hours of labelled speech segments for use in acoustic model training. Additionally, we used 127 hours of out-of-domain data from the AMI Corpus of multi-party meetings[1] using a setup based on [14]. This data is not in general well-matched to the TED-domain. The OOD data was not used directly in acoustic model training, but used to generate out-of-domain neural network features for the in-domain data.

3.2. Deep neural network systems

For our 2012 system, we used neural networks within the tandem framework [15, 16], using DNNs to generate log probabilities over monophones. The monophone probabilities are decorrelated and projected to 30 dimensions, then augmented with the original acoustic features to give a total feature vector of 69 dimensions. These vectors are used for standard HMM-GMM training. Additionally in this year's system, we used DNNs in a hybrid configuration, generating posterior probabilities over tied-state triphones, as proposed in [17]. These are converted to pseudo-likelihoods for use in the decoder.

Both tandem and hybrid nets used PLP input features with 9 frames of temporal context. For the tandem systems, the final nets used had four hidden layers with 1024 hidden units per layer; the hybrid systems used six hidden layers with 2048 hidden units per layer. The tandem nets had an output layer of size 46; the size of the output layer of the hybrid nets varies according to the number of tied states, which resulting from clustering with a GMM; it was typically around 6,000. The nets were trained with a tool based on the Theano library [18] on NVIDIA GeForce GTX 690 GPUs. For the tandem systems, we applied speaker adaptive training of the GMMs using CMLLR [19] regression class trees with 32 classes. For the hybrid systems, we performed adaptation of the input feature space at training and test time using a global CMLLR transform for each speaker. Tandem systems were discriminatively trained with MPE.

As in the 2012 system, we incorporated out-of-domain data using the MLAN technique. Neural networks were trained on the AMI corpus and the resulting nets used to generate posterior features for each utterance in the TED corpus. Th2ese neural net features are known to provide a degree of domain-independence [20]. In the MLAN scheme, the OOD features are augmented with the original acoustic features and a further DNN is trained on these features, allowing further adaptation to the target domain. This second adaptive network may be used to generate tandem features, or used in a hybrid system. The possible configurations are illustrated in Figure 1.

3.3. Voice activity detection

The voice activity detection component of the system comprises a GMM-HMM based model which is used to perform a Viterbi decoding of the audio. The HMM has 2 classes: speech and non-speech. These are modelled with diagonal-covariance GMMs with 12 and 5 mixtures respectively. We allow more mixture components for speech to

[1] http://www.amiproject.org/

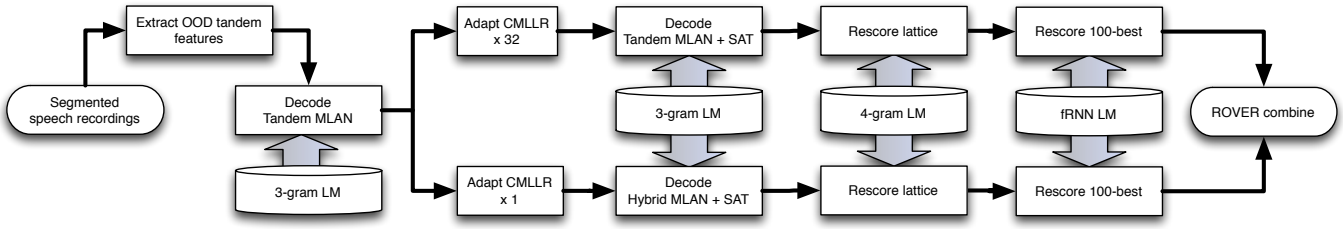


Figure 2: The full decoder architecture

cover its greater variability. Features are calculated every 10ms from a 30ms analysis window and have a dimensionality of 14 (13 PLPs and energy). Models were trained on 70 hours of scenario meetings data from the AMI corpus using the provided manual segmentations as a reference. To avoid over segmentation a minimum duration constraint of 50ms is enforced by inserting a series of 50 states per class that each have a transition weight of 1.0 to the next, the final state has a self transition weight of 0.9.

4. Decoder architecture

Figure 2 shows the complete decoding architecture. After an initial pass, used to generate transcripts to estimate speaker transforms, we operate two parallel decoding sequences for the tandem and hybrid acoustic models. For each model, the complete process consists of a decoding with the trigram LM using HTK's HDecode[2]. Lattices output from the this pass were rescored using the 4-gram LM, generating 100-best lists, which were rescored with the final interpolated RNN LM. Finally, the one-best outputs from tandem and hybrid systems are combined at the hypothesis level using ROVER.

5. Results

In this section we first present development results from individual components of the complete system pipeline. Table 5 shows results using the manual segmentations provided for earlier evaluations. The results may differ slightly from official results due to variations in scoring procedure. It may be observed that there is no clear winner out of the tandem and hybrid systems; however, they are clearly complementary as system combination consistently yields improved performance.

The trends are similar when the automatic segmentation is used, shown in Table 6. When the automatic segmentation is used there is a deterioration in performance of up to 3% WER. Some of this may be attributed to an increase in insertion and deletion errors of the result of segmentation errors; however, an additional source of error, particularly affecting the RNN LM, is that the automatic segmenter typically results in shorter segments, not divided along semantic lines as the manual version is, resulting in reduced language modelling power, since we do not propagate LM probabilities across segment boundaries. Note that the results with the RNN model are available only for a subset of experiments as this component of the system was not fully automatic at the time of system development.

System	dev2010	tst2010	tst2011
Tandem MLAN	15.9	14.1	11.2
+ 4gram	15.6	13.6	10.8
+ RNN	-	-	10.4
Hybrid MLAN	15.6	13.9	11.5
+ 4gram	15.2	13.5	11.3
+ RNN	-	-	10.5
ROVER combination			
4gram	**14.7**	**12.6**	10.3
+ RNN	-	-	**9.9**

Table 5: *Development system results with manual segmentation (WER%)*

System	dev2010	tst2010	tst2011
Tandem MLAN	18.8	17.6	14.9
+ 4gram	18.4	17.2	14.5
+ RNN	17.6	16.6	-
Hybrid MLAN	18.6	17.4	14.6
+ 4gram	18.4	17.2	14.3
+ RNN	17.6	16.7	-
ROVER combination			
4gram	17.6	16.2	**13.2**
+ RNN	**17.0**	**16.1**	-

Table 6: *Development system results with automatic segmentation (WER%)*

Finally, we provide the official results from the 2013 evaluation in Table 7. Automatic segmentation is used only for `tst2013` set. It is notable that the WER is substantially higher on this set than on the other development and evaluation sets. A preliminary analysis suggests that this is probably not due to problems with the segmentation, as insertion and deletion errors do not make up a noticeably higher proportion of the total errors than for the other test sets. Over the talks, the WER ranges from 9% to 48%, suggesting that

[2] `http://htk.eng.cam.ac.uk`

	tst2011	tst2012	tst2013
Primary system	10.2	11.6	22.1

Table 7: *Official system results from the 2013 evaluation (WER%)*

perhaps this year's test set contains a more diverse range of acoustic conditions.

6. Machine translation

We applied machine translation to the ASR output. Details may be found in the accompanying paper [2]. Table 8 compares MT performance for various inputs from the ASR system. Note that performing translation from a confusion network containing multiple ASR hypotheses resulted in worse results that using the one-best output. We are investigating the reasons for this – one theory is that, due to the generally low WER of the systems, the alternative hypotheses are rarely correct, often simply indicating OOV errors when they have high acoustic scores. Table 9 presents, for reference, the official 2013 BLEU results comparing, as inputs, the use of our best system, and the transcription by the IWSLT organisers.

ASR input	en-fr
1-best	22.9
1-best punctuated	24.1
Confusion net	18.4

Table 8: Cased BLEU results for models when tuned and tested on ASR output in different formats.

	en-fr
Edinburgh ASR system	22.45
IWSLT ASR system	23.00

Table 9: Official test 2013 cased BLEU results for 1Best SLT input. The Edinburgh ASR system input was our primary system.

7. Conclusions

We have described our ASR system for the English 2013 IWSLT evaluation Improvements to our system since the 2012 evaluation result in relative WER reductions of 17% 19% on the `tst2011` and `tst2012` sets respectively. The use of RNN LMs does not give improved performance on the `tst2013` set, a result that is probably due to the shorter utterances derived from the automatic segmentation.

Improvements planned for future systems include the use of neural network based voice activity detection, and the pooling of German and English audio data in multi-condition DNN training, whereby both systems are trained simultaneously, sharing lower layers of the network. We also plan to apply talk-level language model adaptation.

8. References

[1] J. Driesen, P. Bell, and S. Renals, "Description of the UEDIN system for german ASR," in *Proc. IWSLT*, 2013.

[2] A. Birch, N. Durrani, and P. Koehn, "Edinburgh SLT and MT system description for the IWSLT 2013 evaluation," in *Proc. IWSLT*, Heidelberg, Germany, 2013.

[3] R. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proc. ACL Conference Short Papers*, Uppsala, 2010, pp. 220–224.

[4] H. Yamamoto, Y. Wu, C. Huang, X. Lu, P. Dixon, S. Matsuda, C. Hori, and H. Kashioka, "The NICT ASR system for IWSLT 2012," in *Proc. International Workshop on Spoken Language Translation*, Hong Kong, Dec. 2012.

[5] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *Proc. ICSLP*, vol. 2, 2002, pp. 901–904.

[6] E. Hasler, P. Bell, A. Ghoshal, B. Haddow, P. Koehn, F. McInnes, S. Renals, and P. Swietojanski, "The UEDIN systems for the IWSLT 2012 evaluation," in *Proc. International Workshop on Spoken Language Translation*, Hong Kong, Dec. 2012.

[7] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, pp. 1137–1155, 2003.

[8] H. Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[9] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*. ISCA, 2010, pp. 1045–1048.

[10] T. Mikolov, M. Karafiát, L. Burget, J. Černokcý, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, 2010.

[11] P. Bell, H. Yamamoto, P. Swietojanski, Y. Wu, F. McInnes, C. Hori, and S. Renals, "A lecture transcription system combining neural network acoustic and language models," in *Proc. Interspeech*, Lyon, France, Aug. 2013.

[12] P. Bell, M. Gales, P. Lanchantin, X. Liu, Y. Long, S. Renals, P. Swietojanski, and P. Woodland, "Transcription of multi-genre media archives using out-of-domain

data," in *Proc. IEEE Workshop on Spoken Language Technology*, Miama, Florida, USA, Dec. 2012.

[13] A. Stan, P. Bell, and S. King, "A grapheme-based method for automatic alignment of speech and text data," in *Proc. IEEE Workshop on Spoken Language Technology*, Miama, Florida, USA, Dec. 2012.

[14] T. Hain, L. Burget, J. Dines, P. Garner, F. Grezl, A. Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "Transcribing meetings with the AMIDA systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 486–498, 2012.

[15] H. Hermanksy, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000, pp. 1635–1630.

[16] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "On using MLP features in LVCSR," in *Proc. Interspeech*, 2004.

[17] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[18] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proc. SciPy*, June 2010.

[19] M. Gales, "Maximum likelihood linear transforms for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 75-98, 1998.

[20] A. Stolcke, F. Grézl, M.-Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in *Proc. ICASSP*, 2006.

The NAIST English Speech Recognition System for IWSLT 2013

Sakriani Sakti, Keigo Kubo, Graham Neubig, Tomoki Toda, Satoshi Nakamura

Augmented Human Communication Laboratory,
Graduate School of Information Science,
Nara Institute of Science and Technology, Japan
{ssakti,keigo-k,neubig,tomoki,s-nakamura}@is.naist.jp

Abstract

This paper describes the NAIST English speech recognition system for the IWSLT 2013 Evaluation Campaign. In particular, we participated in the ASR track of the IWSLT TED task. Last year, we participated in collaboration with Karlsruhe Institute of Technology (KIT). This year is our first time to build a full-fledged ASR system for IWSLT solely developed by NAIST. Our final system utilizes weighted finite-state transducers with four-gram language models. The hypothesis selection is based on the principle of system combination. On the IWSLT official test set our system introduced in this work achieves a WER of 9.1% for tst2011, 10.0% for tst2012, and 16.2% for the new tst2013.

1. Introduction

Similar to the IWSLT 2012 Evaluation Campaign [1], IWSLT 2013 featured an Automatic Speech Recognition (ASR) track in which systems are required to recognize the recordings made available by TED on their website[1]. TED talks bring together the world's most fascinating thinkers and doers, who are challenged to give the talk of their lives in about 5-25 minutes covering topics related to technology, entertainment and design (TED). Spanning everything, from internet trends to solving the world's water supply problems, today TED is a global movement "riveting talks by remarkable people free to the world".

This paper describes the NAIST English speech recognition system. The main challenge of this ASR track is to develop a system that is capable of recognizing spontaneous and open-domain speeches. Last year, we participated in collaboration with Karlsruhe Institute of Technology (KIT). This year is our first time to build a full-fledged ASR system for IWSLT solely developed by NAIST. Our system utilizes weighted finite-state transducers which is based on the Kaldi speech recognition toolkit [2]. Basically, our strategy in this year is to explore and investigate various acoustic features (MFCC, PLP, FBANK), front-end processing (LDA, STC, fMLLR, SAT), and acoustic models (HMM/GMM, SGMM, DNN) provided in the Kaldi toolkit, as well as various grapheme-to-phoneme strategy (Sequitur G2P, DirectTL+, Structured ARROW). However, in case of language models, only traditional four-gram language models were performed at the moment.

The final submission is based on the principle of system combination. The underlying assumption of system combination is that different systems commit different errors which may cancel each other out. However due to limited time, we were not able to submit the full-set combination system. The submitted system was a combination of (HMM/GMM MFCC + SGMM MFCC + HMM/GMM FBANK + SGMM FBANK + DNN FBANK). Furthermore, only half of data were used to train DNN-FBANK model. Nevertheless, experiment results reveal that in comparison with last year best system on the 2011 and 2012 evaluations set which serves as a progress test set, we were still able to reduce the word error rate of our transcription systems from 10.9% to 9.1% for tst2011 and from 12.1% to 10.0% for tst2012, giving a relative reduction of 16.5% and 17.4% respectively. And on the new official 2013 evaluation set, the final system reached a WER of 16.2%.

The rest of this paper is structured as follows. Section 2 summarizes data resources used for the experiments, and Section 3 provides a description of acoustic front-ends used in our system. An overview of the techniques and data used to build our acoustic models is given in Section 4. We describe the vocabulary and language model used for this evaluation in Section 5 and pronunciation lexicon including grapheme-to-phoneme conversion in Section 6. Our decoding strategy and experimental results are explained in Section 7. Finally, the conclusion is drawn in Section 8.

2. Data Resources

2.1. Training Corpora

For acoustic model training, we used TED talks released before the cut-off date of 31 December 2010, downloaded from the TED websites with the corresponding subtitles. The collected talk resulting in a total of 157 hours of speech.

For language model training, the following text corpora provided by the IWSLT organizer were used:

- 2M words of TED transcripts.
- The English portion of the English-French training data from the Sixth Workshop on Statistical Machine

[1] http://www.ted.com/talks

Translation (WMT 2011), including EuroParl (EPPS), News Commentary (NC), and NEWS.

Table 1: Total size (word count) and vocabulary size of the individual text corpora.

Data	Size	Vocabulary
TED	2.7m	45k
EPPS	54m	82k
NC	4.5m	50k
NEWS	2,402m	1,047k

We normalized the text corpora of TED, EPPS, NC, and NEWS, in a case-insensitive fashion. Table 1 shows the resulting text corpora along with their total size (word count) and vocabulary size.

2.2. Test Corpora

Concerning the test corpora, the development and evaluation data sets (dev2010, tst2010, dev2012) used in past editions, were provided by IWSLT organizer for development purposes. As for evaluation purposes, evaluation data sets of tst2011 and tst2012 were used as the progress test set to compare the results of this year against the best results achieved in 2011 and 2012. Then, a new released test set for official test set of 2013 (tst2013) were used for final evaluation of our systems.

3. Front-End Processing

We investigated the use of three different kinds of acoustic front-ends: (1) mel-frequency cepstral coefficients (MFCC), (2) perceptual linear prediction (PLP)[3] and (3) log mel-filter bank (FBANK). The frontend provides features every 10ms with 25ms width. For each utterance in the speech training data, 13 static of acoustic features (MFCCs, PLPs, or FBANKs) including zeroth order for each frame are extracted and normalized with cepstrum mean normalization in order to have zero mean per speaker.

To incorporate the temporal structures and dependencies, 9 adjacent frames (4 frames on each left-right side of the current frame) of the acoustic features (MFCCs, PLPs, or FBANKs) are spliced together into one single feature vector leading to 117 dimensional super vectors (9x13 dimensions). These are then projected down to an optimum 40 dimensions by applying a linear discriminant analysis (LDA). After that, the resulting features are further de-correlated using maximum likelihood linear transformation (MLLT)[4], which is also known as global semi-tied covariance (STC)[5] transform. Moreover, speaker adaptive training (SAT)[6] is performed using a single feature-space maximum likelihood linear regression (fMLLR)[7] transform estimated per speaker.

4. Acoustic Model

Acoustic models are trained on the LDA+STC+fMLLR features describe above. We employed 39 phonemes of English based on CMU dictionary without stress information. Additionally, we added 9 special phoneme of non-speech sounds derived from TED speech sources. These include *SIL* for silence, *SENTSTART* and *SENTEND* for head and tail TED's sound effect, and *APPLAUSE*, *BEEP*, *LAUGHTER*, *MUSIC*, *NOISE*, and *VOICENOISE* for sound that appeared in TED speech sources.

Here we investigated the use three different kinds of acoustic models: (1) Hidden Markov Model/Gaussian Mixture Model (HMM/GMM) (2) Subspace Gaussian Mixture Models (SGMM) (3) Deep Neural Network (DNN) which are described below.

- **HMM/GMM**
  Three-state left-to-right HMM topology without skip states. The HMM units are derived from 39 phonemes of English. Each phoneme is classified by its position in word (4 classes: begin, end, internal and singleton). Context-dependent cross-word triphone HMMs were first trained with GMM output probability. The final model totally include 320K Gaussians trained with boosted maximum mutual information (MMI)[8] criterion of discriminative training.

- **SGMM**
  For SGMM, the Kaldi toolkits provides an implementation of the approach described in [9]. In this case, HMMs are builts with subspace GMM output probability. The final model consists of 9.1K states, which is also trained with boosted maximum mutual information (MMI) [8] criterion of discriminative training.

- **DNN**
  Here, we performed HMM/DNN hybrid framework, in which the network is trained with 7 layers, where each hidden layer has 2048 neurons. This DNN is initialized with stacked restricted Boltzmann machines (RBMs) that are pretrained in a greedy layerwise fashion.

5. Vocabulary and Language Model

5.1. Vocabulary

For the vocabulary selection, we followed an approach proposed by Venkataraman et al. [10]. We built unigram language models from all text sources, and combined them to satisfy unigram probabilities that maximize the likelihood of a held-out TED data set dev2010, by using the SRILM toolkit [11]. We then defined the 100k most probable unigrams from the combined unigram language model as the vocabulary.

5.2. LM Training

We constructed a 3-gram language model for decoding a utterance, and a 4-gram language model for rescoring hypothe-

ses. At first, we built 3-gram and 4-gram language models with modified Kneser-Ney smoothing [12] from each of the text corpora by using kaldi LM toolkit[2]. These were then combined per n-gram language model using linear interpolation as follows:

$$P(w|h) = \lambda_1 P_1(w|h) + \lambda_2 P_2(w|h) + \cdots + \lambda_k P_k(w|h) \quad (1)$$

The interpolation weights $\lambda_1, \ldots, \lambda_k$ were chosen to maximize the likelihood of a held-out TED data set dev2010. Additionally, we pruned the n-gram entries that have a lower probability than 5e-10 in the combined 3-gram language model. For combining and pruning the language model, we employed the SRILM toolkit. The combined and pruned 3-gram language model contains 20 million bigrams, 45 million trigrams. The combined 4-gram language model contains 35 million bigrams, 194 million trigrams, and 397 million 4-grams. Perplexities on tst2010 for each 3-gram and 4-gram language model is shown in Table 2.

Table 2: Language model perplexities on tst2010 for each 3-gram and 4-gram language models. The n-gram entries that have a lower probability than 5e-10 in the 3-gram language model is pruned.

Data	3-gram	4-gram
TED	174.38	170.82
EPPS	450.38	429.14
NC	413.97	410.51
NEWS	200.63	192.30
Combined	138.58	127.72

6. Dictionary

6.1. G2P conversion

G2P conversion is employed to obtain a pronunciation of words that does not exist in a dictionary. We try three G2P conversion methods, (1) joint n-gram model [13] as implemented in Sequitur G2P (*Sequitur*), (2) DirecTL+ (*DirecTL+*) which is an online discriminative training based on MIRA for G2P conversion [14, 15] and (3) Structured AROW [16] which is also an online discriminative training that extends AROW [17] to structured learning (*SAROW*).

Table 3: Phoneme error rate (PER), word error rate (WER) and learning time (Time) for each G2P conversion methods in the CMU dictionary.

	PER(%)	WER(%)	Time(hr.)
Sequitur	6.77	28.55	17.5
DirecTL+	6.19	26.38	55.4
SAROW	6.15	26.48	28.5

[2]http://merlin.fit.vutbr.cz/kaldi/kaldi_lm.tar.gz

We have compared these methods in a preliminary experiment in term of phoneme error rate (PER) and word error rate (WER). In the CMU dictionary, we have employed 10% as test data, 5% as development data and the reminder as training data. As showing in Table 3, *DirecTL+* and *SAROW* significantly improved over *Sequitur* in terms of PER and WER. The *SAROW* was almost the same performance as the *DirecTL+* in terms of PER and WER, while the *SAROW* improved the learning time of the *DirecTL+*. From the learning time of the *SAROW*, We determined to employ Structured AROW as our G2P conversion method in dictionary construction.

6.2. Dictionary construction

We first constructed a G2P model with Structured AROW as described above. Here, all data in the CMU dictionary were employed as training data. For some training parameters such as learning iteration, we re-used parameters employed in the preliminary experiment. After that, we applied the trained G2P model to a word that appears in the language model but does not appear in the CMU dictionary, except abbreviation words with all capitalized letters. The pronunciation of abbreviation words were constructed based on rule in which in each letter is converted to the corresponding single-letter pronunciation. The number of the converted word was 36k words in the 100k vocabulary.

7. Decoding Strategy and Results

Our decoding algorithms use weighted finite state transducers (WFSTs)[18] based on Kaldi speech recognition toolkit[2], a free, open-source toolkit for speech recognition research. The decoding-graph construction process is basically based on the conventional recipe described in [18] with slight modification to allow different phones to share the same context-dependent states.

7.1. Single System

Figure 1 shows the results given various configurations on the use of different acoustic features and acoustic models. For comparison we evaluated the performance on the development set. The results reveal that on each development set, DNN models with MFCCs, PLPs, or FBanks always outperformed HMM/GMM and SGMM. On the most left "dev2010" is the ASR performance on development set of 2010 given the segmentation data, while on the second one "dev2010 (no seg)" is the ASR performance on development set without time segmentation information. As can be seen, without time segmentation, the performance of ASR systems slightly reduced.

7.2. Combination System

Here, we investigate model combination system, feature combination system and full combination described below.

- **Model Combination System**
  Here, we focus to investigate the ASR performance

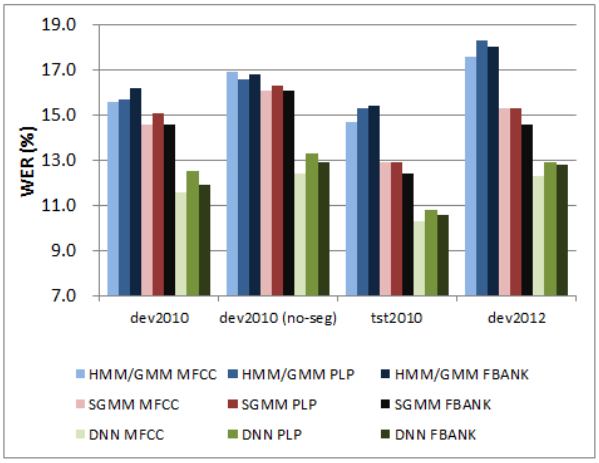


Figure 1: Performance of the single system on the development set and test set in WER.

of each acoustic features of MFCCs, PLPs, and FBANKs. Figure 2 shows the results of those acoustic features in combination of all acoustic models (HMM/GMM+SGMM+DNN). In average, the performance of those features are mainly the same. In most systems, the combination with optimum weight provide an improvement of the performances. Unfortunately, MFCC (HMM/GMM+SGMM+DNN) combination system performed worse than the best MFCC (DNN) single system. This is because the optimum weight was calculated at once (globally) based on the results of all single systems in all development sets, which may not be effective for all cases.

- **Feature Combination System**
  Here, we focus to investigate the ASR performance of each acoustic models of HMM/GMM, SGMM, and DNN. Figure 3 shows the results of those acoustic models in combination of all acoustic features (MFCCs+PLPs+FBANKs). The HMM/GMM always performed the worst. The best performance was achieve with DNN. However, the combination with optimum weight does not provide any significant improvement of the performances.

- **Full Combination System**
  Here, we perform feature and model combination system from 4-combination system to the full 9-combination system. Figure 4 shows the results of those acoustic models in combination of various acoustic features (MFCCs+PLPs+FBANKs) and various acoustic models (HMM/GMM+SGMM+DNN). The results reveal that the full 9-combination system provide a better performance than others. However, it is quite surprising that there is no significant improvement from 4-combination system to the full 9-combination system.

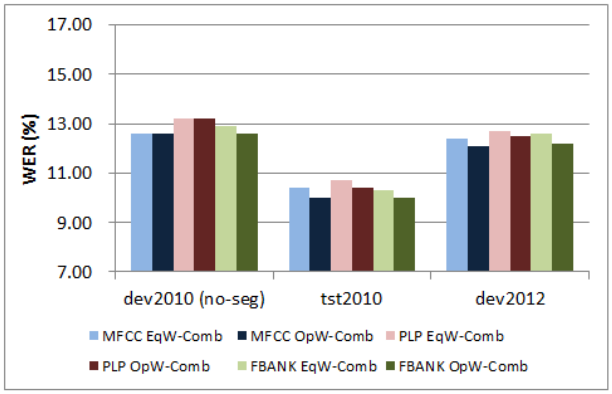


Figure 2: Performance of each acoustic features of MFCCs, PLPs, and FBANKs with acoustic model combination (HMM/GMM+SGMM+DNN) on the development set in WER.

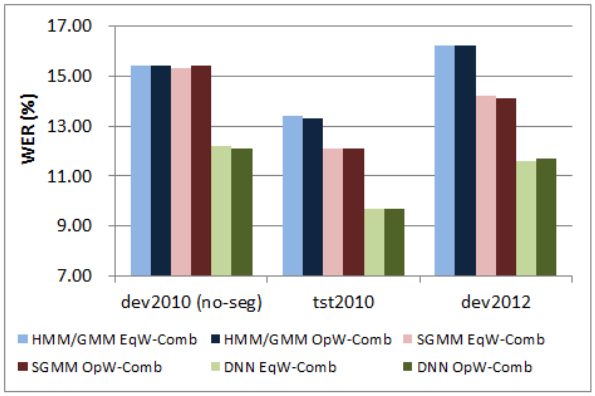


Figure 3: Performance of each acoustic models of HMM/GMM, SGMM, and DNN with acoustic features combination (MFCCs+PLPs+FBANK) on the development set and test set in WER.

7.3. Final Submission System

As we described previously, due to a limited time, we were not able to submit the full-set of 9-combination system. Our submitted primary system was based on a combination of (HMM/GMM MFCC + SGMM MFCC + HMM/GMM FBANK + SGMM FBANK + DNN FBANK). Table 4 shows the summary of our final system based on IWSLT 2013 evaluation feedback in comparison with the best system from feature combination, model combination, and the full 9-combination system.

In comparison with last year best system, experiment results reveal that the performance of the submitted system on the 2011 and 2012 evaluations set which serves as a progress test set, were still able to reduce the word error rate of our transcription systems from 10.9% to 9.1% for tst2011 and from 12.1% to 10.0% for tst2012, giving a relative reduction of 16.5% and 17.4% respectively. And on the new official 2013 evaluation set, the submitted system reached a WER of

16.2%. However, the best performance was provide by full 9-combination system which reached a WER of 15.6% giving another 3.7% relative reduction from the submitted system.

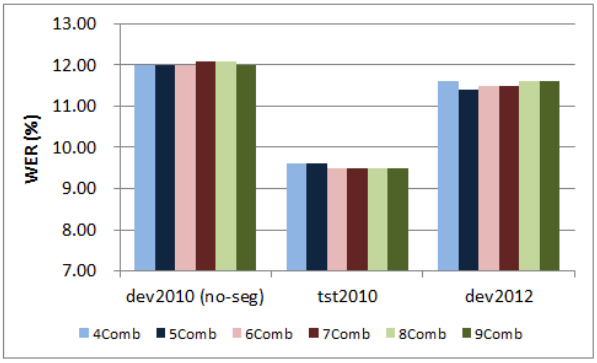


Figure 4: Performance of 4-combination system to the full 9-combination system on the development set and test set in WER.

ASR System	tst2011	tst2012	tst2013
Model Combination System	9.4%	10.4%	16.1%
Feature Combination System	9.2%	10.1%	16.0%
Full 9-Combination System	9.0%	9.7%	15.6%
Official Submitted System	9.1%	10.0%	16.2%

Table 4: Summary of final system performances based on IWSLT 2013 evaluation feedback in comparison with the best system from feature combination, model combination, and the full 9-combination system.

8. Conclusion

In this paper we described our English speech-to-text system with which we participated in the IWSLT 2013 TED task evaluation on the ASR track. The decoding strategy for the final submission is based on the principle of system combination. The underlying assumption of system combination is that different systems commit different errors which may cancel each other out. However due to a limited time, we are not able to submit the full-set combination system. The submitted system was a combination of (HMM/GMM MFCC + SGMM MFCC + HMM/GMM FBANK + SGMM FBANK + DNN FBANK). Nevertheless, experiment results reveal that on the 2011 and 2012 evaluations set which serves as a progress test set, we were still able to reduce the word error rate of our transcription systems from 10.9% to 9.1% for tst2011 and from 12.1% to 10.0% for tst2012, giving a relative reduction of 16.5% and 17.4% respectively. And on the new official 2013 evaluation set, the final system reached a WER of 16.2%. The best performance was provided by full 9-combination system which reached a WER of 15.6% giving another 3.7% relative reduction from the submitted system.

9. References

[1] M. Federico, L. Bentivogli, M. Paul, and S. Stueker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc. IWSLT 2012*, Hong Kong, 2012, pp. 12–33.

[2] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Moticek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, Hawaii, USA, 2011.

[3] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1998.

[4] R. Gopinath, "Maximum likelihood modeling with gaussian distributions for classification," in *Proc. of ICASSP*, 1998, pp. 661–664.

[5] M. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[6] R. S. T. Anastasakos, J. Mcdonough and J. Makhoul, "A compact model for speaker adaptive training," in *Proc. ICSLP*, 1996, pp. 1137–1140.

[7] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[8] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. of ICASSP*, Las Vegas, USA, 2008, pp. 4057–4060.

[9] D. Povey, L. B. D. Povey, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiat, A. Rastrow, R. Rose, P. Schwarz, and S. Thomas, "The subspace Gaussian mixture model – a structured model for speech recognition," in *Proc. of ICASSP*, Las Vegas, USA, 2008, pp. 4057–4060.

[10] A. Venkataraman and W. Wang, "Techniques for effective vocabulary selection," in *Proc. of EUROSPEECH*, Geneva, Switzerland, 2003, pp. 245–248.

[11] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. of ICSLP*, Denver, USA, 2002, pp. 901–904.

[12] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proc. ICASSP*, 1995, pp. 181–184.

[13] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[14] S. Jiampojamarn and G. Kondrak, "Online discriminative training for grapheme-to-phoneme conversion," in *Proc. INTERSPEECH*, Beijing, China, 2009, pp. 1303–1306.

[15] C. C. S. Jiampojamarn and G. Kondrak, "Integrating joint n-gram features into a discriminative training framework," in *Proc. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, Beijing, China, 2010, pp. 697–700.

[16] K. Kubo, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "Grapheme-to-phoneme conversion based on adaptive regularization of weight vectors," in *Proc. INTERSPEECH*, 2013, pp. 1946–1950.

[17] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," *Advances In Neural Information Processing Systems*, pp. 414–422, 2009.

[18] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech and Language*, vol. 20, no. 1, pp. 69–88, 2002.

The KIT Translation Systems for IWSLT 2013

Thanh-Le Ha, Teresa Herrmann, Jan Niehues, Mohammed Mediani,
Eunah Cho, Yuqi Zhang, Isabel Slawik and Alex Waibel

Institute for Anthropomatics
KIT - Karlsruhe Institute of Technology
firstname.lastname@kit.edu

Abstract

In this paper, we present the KIT systems participating in all three official directions, namely English→German, German→English, and English→French, in translation tasks of the IWSLT 2013 machine translation evaluation. Additionally, we present the results for our submissions to the optional directions English→Chinese and English→Arabic.

We used phrase-based translation systems to generate the translations. This year, we focused on adapting the systems towards ASR input. Furthermore, we investigated different reordering models as well as an extended discriminative word lexicon. Finally, we added a data selection approach for domain adaptation.

1. Introduction

In the IWSLT 2013 Evaluation Campaign [1], we participated in the tasks for text and speech translation for all the official language pairs: English→German, German→English and English→French as well as two optional directions. The TED tasks consist of automatic translation of both the manual transcripts (MT task) and transcripts generated by automatic speech recognizers (SLT task) for talks held at the TED conferences[1]. For German→English, the test data was collected from the TEDx project[2].

The TED talks are given in English in a large number of different domains. Some of these talks are manually transcribed and translated by global volunteers into many languages [2]. The TED translation tasks this year bring up interesting challenges: (1) the problem of adapting general models - mainly trained on news data - towards the diverse topics in TED talks, (2) the need of universal techniques for translating texts from and to various languages, and (3) the appropriate solution for inserting punctuation marks and case information on automatic speech recognition (ASR) outputs for the spoken language translation (SLT) task.

To deal with those challenges, we provided several advanced adaptation methods both for translation and language models to leverage both the wide coverage of large data portions and the domain-relevance of the TED corpus. In addition, we optimized our universal techniques to better conform with different languages.

Compared to our last year's system, we focused on four new components: handling of ASR input (Section 3), combination of reordering models of different linguistic abstraction levels (Section 4), data selection for language model (LM) adaptation (Section 5) and an extended discriminative word lexicon (Section 6).

The next section briefly describes our baseline system, while Sections 3 through 7 present the different components and extensions used by our phrase-based translation system. After that, the results of the different experiments, including official and optional language pair systems, are presented and we close the paper with a conclusion.

2. Baseline System

Among the parallel data provided, we utilize EPPS, NC, TED, Common Crawl for English→German and German→English, plus Giga for English→French. The monolingual data we used include the monolingual part of those parallel data, the News Shuffle corpus for all three directions and additionally the Gigaword corpus for English→French and German→English.

A common preprocessing is applied to the raw data before performing any model training. This includes removing long sentences and sentences with length difference exceeding a certain threshold. In addition, special symbols, dates and numbers are normalized. The first letter of every sentence is smart-cased. In German→English, we also apply compound splitting [3] to the source side of the corpus. Furthermore, an SVM classifier is used to filter out the noisy sentence pairs in the Giga English→French corpus and the Common Crawl as described in [4].

Unless stated otherwise, the language models used are 4-gram language models with modified Kneser-Ney smoothing, trained with the SRILM toolkit [5] and scored in the decoding process with KenLM [6]. The word alignment of the parallel corpora is generated using the GIZA++ Toolkit [7] for both directions. Afterwards, the alignments are combined using the grow-diag-final-and heuristic. For German→English, we use a discriminative word alignment (DWA) approach [8]. The phrases are extracted using the

[1] http://www.ted.com
[2] http://www.ted.com/tedx

Moses toolkit [9] and then scored by our in-house parallel phrase scorer [10]. Phrase pair probabilities are computed using modified Kneser-Ney smoothing as in [11].

In all directions, beside the word-based language models, some of the non-word language models are used. In order to increase the bilingual context used during the translation process, we use a bilingual language model as described in [12]. To model the dependencies between source and target words even beyond borders of phrase pairs, we create a bilingual token out of every target word and all its aligned source words. The tokens are ordered like the target words. In addition, to alleviate the sparsity problem for surface words, we use a cluster language model based on word classes. This is done in the following way: In a first step, we cluster the words of the corpus using the MKCLS algorithm [13]. Then we replace the words in the TED corpus by their cluster IDs and train an n-gram language model on this corpus consisting of word classes.

3. Preprocessing for Speech Translation

The system translating automatic transcripts needs special preprocessing on the data, since generally there is no or no reliable case information and punctuation in the automatically generated transcripts. We have used a monolingual translation system as shown in [14] to deal with the difference in casing and punctuation between a machine translation (MT) and an SLT system. In contrast to the condition in their work, in this evaluation campaign sentence boundaries are present in the test sets. Therefore, we use this monolingual translation system for predicting commas instead of all punctuation marks in the test set. In addition to predicting commas, we also predict casing of words using the monolingual translation system. This preprocessing will be denoted as Monolingual Comma and Case Insertion (MCCI).

In order to build the monolingual system which translates a source language into the same language with commas inserted, we prepare the parallel corpus for training. For the source side of the corpus, we take the preprocessed monolingual corpus of a normal translation system, remove all punctuation marks, and insert a period mark at the end of each line. For the target side of the corpus, we take the preprocessed corpus of same language from the normal translation system and replace all sentence-final punctuation marks such as "!", "?", "." by a period. Therefore, the only difference between the source and the target side corpus is inserted commas on the target side.

In this evaluation campaign we work with two source languages, English and German. Therefore, we build a monolingual translation systems each for the two languages. The speech translation system with English on the source side is built using true-cased English source and target side. As the test set often contains only lower-cased letters, in the English monolingual system we take this already lower-cased, preprocessed automatic transcript for translation. In order to match this input during decoding, the source side of a phrase table is lower-cased. As the case information contains more information for German, the German monolingual translation system is built using lower-cased German source and true-cased target side. All words in the preprocessed German automatic transcript are lowercased, but are translated into true-cased text using the monolingual translation system.

The monolingual translation systems for both languages are built on the corresponding side of the EPPS, TED, and NC corpus, which sum up to 2.2 million sentences. A 4-gram language model trained on the word tokens is used. Word reordering is ignored in these systems. In order to capture more context, we use a 9-gram language model trained on part-of-speech (POS) tokens. Moreover, a 9-gram cluster language model is trained on 1,000 clusters, based on the MKCLS algorithm as described in the baseline system.

For the speech translation tasks, the output of the monolingual translation system becomes the input to our regular translation system which is trained using data with punctuation marks.

4. Word Reordering Model

Word reordering is modeled in two ways. The first is a lexicalized reordering model [15] which stores reordering probabilities for each phrase pair. The second model consists of automatically learned rules based on POS sequences and syntactic parse tree constituents and performs source sentence reordering according to target language word order.

The rules are learned from a parallel corpus with POS tags [16] for the source side and a word alignment to learn continuous reordering rules that cover short-range reorderings [17]. Discontinuous rules consist of POS sequences with placeholders and allow long-range reorderings [18]. In addition, we apply a tree-based reordering model [19] to better address the differences in word order between German and English. Syntactic parse trees [20, 21] for the source side of the training corpus and a word alignment are required to learn rules on how to reorder the constituents in the source sentence to simulate target sentence word order. The POS-based and tree-based reordering rules are applied to each input sentence before translation. The resulting reordered sentence variants as well as the original sentence are encoded in a word lattice.

In order to apply the lexicalized reordering model, the lattice includes the original position of each word. Then the lattice is used as input to the decoder. During decoding the lexicalized reordering model provides the reordering probability for each phrase pair. At the phrase boundaries, the reordering orientation with respect to the original position of the words is checked. The probability for the respective orientation is included as an additional score in the log-linear model of the translation system.

5. Adaptation

In order to achieve the best performance on the target domain, we perform adaptation for translation models as well as language models.

We adapt the translation model (TM) by using the scores from the in-domain and out-of-domain phrase table as described in the backoff approach [22]. This results in a phrase table with six scores, the four scores from the general phrase table as well as the two conditional probabilities from the in-domain phrase table. In addition, we adapt the candidate selection in some of our systems by taking the union of the candidates translations from both phrase tables (CSUnion).

The language model (LM) is adapted by log-linearly combining the general language model and an in-domain language model trained only on the TED data. In addition, in some of the systems we combine these language models with a third language model. This language model was trained on data automatically selected using cross-entropy differences [23]. We selected the top 5M sentences to train the language model.

6. Discriminative Word Lexica

Mauser et al. [24] have shown that the use of DWL can improve the translation quality. For every target word, they train a maximum entropy model to determine whether this target word should be in the translated sentence or not using one feature per source word. In our system we use the extended version using also source context and target context features [25]. When using source context features, not only the words of the sentence are used as features, but also the n-grams occurring in the sentence. The target context features encode information about the surrounding target words.

One specialty of the TED translation task is that we have a lot of parallel data we can train our models on. However, only a quite small portion of these data, the TED corpus, is very important for the translation quality. Therefore, we achieve a better translation performance by training the models only on the TED data.

7. Continuous Space Language Model

In recent years, different approaches to integrate continuous space models have shown significant improvements in the translation quality of machine translation systems [26]. Since the long training time is the main disadvantage of this model, we only train it on the small, but very domain-relevant TED corpus.

In contrast to most other approaches, we did not use a feed-forward neural network, but used a Restricted Bolzmann Machine (RBM). The main advantage of this approach is that the free energy of the model, which is proportional to the language model probability, can be calculated very efficiently. Therefore, we are able to use the RBM-based language model during decoding and not only in the rescoring phase.

The RBM used for the language model consists of two layers, which are fully connected. In the input layer, for every word position there are as many nodes as words in the vocabulary. Since we used a 4-gram language model, there are 4 word positions in the input layer. These nodes are connected to 32 hidden units in the hidden layer. The model is described in detail in [27].

8. Results

In this section, we present a summary of our experiments for all tasks we have carried out for the IWSLT 2013 evaluation. All the reported scores are case-sensitive BLEU scores calculated based on the provided development and test sets.

8.1. English→German

We conducted several experiments for English→German translation using the available data. They are summarized in Table 1. The baseline system is a phrase-based translation system using POS-based reordering rules. Preprocessing of the source and target language of the training corpora is performed as described above. Adaptation of the phrase table and language model using the in-domain part of the training data is included, as well as a bilingual language model to increase the source context across phrase boundaries. Finally, the baseline system also includes a cluster-based language model using the clusters automatically generated by the MKCLS toolkit.

System	Dev	Test
Baseline	23.58	23.50
+ Tree-based Rules	23.61	23.87
+ Lexicalized Reordering	23.74	23.93
+ POSLM	23.81	24.14
+ DWL	24.44	24.76
+ Class-based 9-gram LMs	24.19	24.93
+ TargetContext + LM DataSelection	24.24	**25.06**

Table 1: Experiments for English→German (MT)

By adding tree-based reordering rules and a lexicalized reordering model we increase the translation quality by more than 0.4 BLEU points. An additional language model for POS sequences gives another increase of 0.2 BLEU points. A remarkable improvement of 0.6 can be observed by introducing a discriminative word lexicon trained on the in-domain data where bigrams are used to include more information about the context words on the source side. Extending the class-based language model to 9-grams leads to further improvement by 0.2. The final system includes target context features in the discriminative word lexicon and a language model trained on 5 million sentences selected from all data based on cross entropy similarity.

8.1.1. SLT Task

For the English→German SLT task, we used one of the systems developed for the MT task. For reordering, it includes the lexicalized reordering model and long-range reordering rules. The tree-based rules are excluded since they do not conform well with the speech data. In addition, the system uses 9-gram POS-based and MKCLS language models and an in-domain DWL with source context. This system ignores case information on the source side. While both development and test data were available for the MT task, for the SLT task only one data set was provided. Therefore, we used it for testing and performed optimization on text data.

In order to adapt the system further towards the task of translating speech input, we added the monolingual comma and case insertion model, which performs a preprocessing step consisting of monolingual translation of lowercased English speech into true-cased English while also inserting commas. For this, no new optimization was performed, only the input was changed. This special treatment of the speech input helped improve the system performance by 1.3 BLEU points. Table 2 shows the overview of the speech translation system.

ASR Adaptation	Test
Baseline	17.60
MCCI	**18.92**

Table 2: Experiments for English→German (SLT)

8.2. German-English

We summarize the development of the German→English system in Table 3. The translation model of the baseline system uses a bilingual language model. It uses all types of reordering rules and a lexicalized reordering model. Furthermore, three language models are combined log-linearly in this system. One language model is trained on all data, one only on the in-domain data and we use one cluster language model trained on all data using 1,000 clusters. Adding the DWL trained on the TED corpus using source and target context features improves the performance by 0.9 BLEU points. Further improvements are achieved by adding a language model trained on the automatically selected data. We further adapt the system to the TED task using the union candidate selection and by adding a RBM-based language model. This improves the system only slightly by 0.1 BLEU points. Finally, we replace the cluster language model by one trained only on the TED corpus and also use morphological operations to translate unknown word forms [12].

8.2.1. SLT Task

For the SLT task, we use the MT system without the in-domain cluster LM and morphological operations. By directly using the MT system to translate the ASR output, a translation quality of 18.33 BLEU points is reached. As there are often no case information and commas in the ASR output, we remove these information from the source side of the phrase table. Using this system, we improve the translation quality to 19.09. Then we use the MCCI system described in Section 3 to insert case information and commas into the ASR output. When translating this modified ASR output, we reach a final BLEU score of 20.1.

System	Dev	Test
Baseline	35.17	29.76
+ DWL	35.42	30.65
+ LM DataSelection	35.51	30.80
+ CSUnion + RBMLM	35.75	30.87
+ In-domain Cluster LM	35.74	31.10
+ Morphological Operations	-	**31.15**

Table 3: Experiments for German→English (MT)

ASR Adaptation	Test
Baseline	18.33
Phrase Table	19.09
MCCI	**20.10**

Table 4: Experiments for German→English (SLT)

8.3. English→French

Table 5 reports some remarkable improvements as we combined several techniques on the English→French direction. The big phrase table is trained on TED, EPPS, NC, Giga and Crawl data, while the language model is trained on the French part of those corpora plus News Shuffle. The system also uses short-range reordering rules derived from smaller data portions (TED, EPPS and NC). The result of this setting is 31.08 BLEU points.

System	Dev	Test
Baseline	27.68	31.08
+ PT+LM Adaptation	28.48	31.76
+ Bilingual LM	28.66	32.57
+ POS+Cluster LMs	28.85	32.53
+ Lexicalized Reordering	29.22	32.83
+ DWL Source Context	29.45	**33.06**

Table 5: Experiments for English→French (MT)

Several advanced adaptations are conducted both on translation and language models. First, the phrase table is adapted using the clean EPPS, NC and TED data. Afterwards, it is adapted towards the TED domain. For the language models, we follow the similar adaptation scheme with the models ranging from in-domain to general-genre data.

We log-linearly combine the language models trained on TED, EPPS, NC, Giga, and Crawl by minimizing the perplexity on the development set. Those adaptation techniques boost the system around 0.7 BLEU points. Further gains come from using different non-word language models. Introducing the bilingual language model leads to a small improvement of 0.18 on Dev and 0.81 BLEU points on Test. Adding a 9-gram POS-based language model and a 4-gram 50-cluster language model trained on in-domain data helps gain almost 0.2 BLEU points on Dev, but results in a slightly reduction of 0.04 on Test. The system is further enhanced by 0.3 BLEU points when we integrated lexicalized reordering probabilities as an independent feature. Finally, by taking the source context of the DWL into account, we achieve the best system with a 0.23 increase, reaching 33.06 BLEU points.

8.3.1. SLT Task

We approached the SLT tasks in two distinct ways. The first is that we use the best system of the MT task to translate the ASR outputs which were already preprocessed by Monolingual Comma and Case Insertion (MCCI) system as mentioned in Section 3. The second approach is the system named ASR-Dedicated, which evolves from rebuilding the translation model from modified Giza alignments dedicated for ASR data only. The modifications consist of removing the case and punctuation marks except the period.

Table 6 presents the results using the best MT system to translate two ASR outputs and from the second approach. The ASR outputs are the raw text without any comma (None) and the output using MCCI preprocessing. The numbers show that a big improvement of almost 3 BLEU points comes from the input preprocessed by MCCI. The commas MCCI inserted have a great effect on the fluency of the ASR output and consequently improved the translation quality. The numbers also show that the system trained and optimized to work best for texts would work adequately for ASR outputs as well.

We submitted the best MT system with MCCI as the primary, and the second approach's result as the contrastive.

ASR Adaptation	Test
None	20.75
MCCI	**23.69**
ASR-Dedicated	22.90

Table 6: Experiments for English→French (SLT)

8.4. English→Arabic

For this pair, we use the parallel data from TED. The UN parallel data is provided in raw format. In order to get useful parallel pairs out of this raw data, we segment the two sides into sentences, exclude all documents having a large difference in number of sentences, sentence-align the resulting document pairs, and finally filter out the noisy sentence pairs.

We use the default sentence segmenter provided by the NLTK toolkit [28] to segment both sides. The sentence alignment is performed using the Hunalign aligner [29]. Since this aligner works better with a lexicon, we build one from Giza alignments trained on the TED corpus. The filtering is carried out using an SVM classifier as stated in Section 2. The tokenization and POS tagging of the Arabic side are performed using the AMIRA toolkit [30].

In addition to the parallel data provided, the fifth edition of the LDC Gigaword Arabic corpus is also used for language modeling.

Table 7 summarizes the experiments for the English→Arabic pair. The baseline translation model is trained on all parallel data (TED and UN) and involves many language models which are log-linearly combined. These include individual models one from each corpus (TED, UN, Gigaword) and two more (UN & TED and all corpora together). In this configuration we use the short range reordering. This system gives 13.15 on Dev and 8.43 on Test. The effect of translation model adaptation is remarkable: it improves the system performance by almost 1.4 BLEU on Dev and 0.26 on Test. Slight improvements could be brought by introducing more language models. For instance, using a bilingual language model trained on all parallel data increases the performance on Dev by almost 0.2 while it has no observable effect on Test. On the other hand, adding a 4-gram cluster language model trained on TED only (with 50 classes) enhances the score on Test by 0.2 while it leaves the Dev score almost unchanged. This last system is used in our submission.

System	Dev	Test
Baseline	13.15	8.43
+ PT Adaptation	14.54	8.69
+ Bilingual LM	14.79	8.70
+ Cluster LM	14.81	**8.92**

Table 7: Experiments for English→Arabic (MT)

8.5. English→Chinese

The English→Chinese system is trained on the bilingual TED and filtered UN corpora. As the UN corpus is document-aligned, we have filtered out about 30k aligned sentences as training data with a KM algorithm. The weight of a sentence pair is the accumulation of word and its translation occurring in a dictionary. The dictionary used here is from LDC (LDC2002L27). The language models are trained on the monolingual TED data and the target side of the whole UN data.

In contrast to European languages, there are no spaces between Chinese words. In our primary system we segment Chinese into characters and tokenize and lowercase English.

Adaptation, reordering and DWL source context models have given contribution to the improvement of translation. In Table 8 we present the steps which achieve improvement. The baseline is a monotone translation with 6-gram language model. As the adaptation described in Section 5, we use the TED corpus as the in-domain data to adapt the phrase table and language model. We use two reordering models: short-range POS-based reordering and lexicalized reordering, which are described in Section 4. Finally, after adding the DWL source context model as described in Section 6 and CSUnion model in Section 5, the BLEU score on test data has gained more than 1 point compared to the baseline.

We have also built a system based on Chinese words as a contrastive system, where the words are generated with the Stanford word segmenter[3].

System	MT		SLT
	Dev	Test	Test
Baseline	14.01	16.75	-
+ Adaptation	14.61	16.77	-
+ POS Reordering	14.71	17.51	-
+ Lexicalized Reordering	14.91	17.18	-
+ DWL+CSUnion	15.14	**17.84**	**17.28**

Table 8: Experiments for English→Chinese

8.5.1. SLT Task

The speech translation system has used the same configuration as the best one for the MT task. We built the test data set by removing the case information and punctuation from the text test data. In order to apply the system trained on text for speech automatic transcripts, we predict commas with the preprocessing described in Section 3. The result is shown in Table 8.

9. Conclusions

In this paper, we presented the systems with which we participated in the TED tasks in both speech translation and text translation of the IWSLT 2013 Evaluation Campaign. Our phrase-based machine translation system was extended with different models.

When translating ASR input, we need to adapt the system to these conditions. Often case information or commas are missing or misplaced. Therefore, we use a method to automatically correct this information in order to directly use our default translation model without training a separate model.

The successful application of different supplementary models trained exclusively on TED data (cluster language model, DWL, and continuous space language model) shows the usefulness and importance of in-domain data for such tasks, regardless of their small size. Furthermore, we could adapt the system even more to the task by using data selection methods.

The DWL allows us to include arbitrary features when calculating the translation probabilities. By extending these models to also include contextual information about the source and target sentence, we were able to increase the translation performance. Furthermore, we could improve the translation performance by combining information about the word order from different linguistic levels.

[3]http://nlp.stanford.edu/software/segmenter.shtml

10. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

11. References

[1] M. Cettolo, J. Niehues, S. Stueker, L. Bentivogli, and M. Federico, "Report on the 10th IWSLT Evaluation Campaign," in *IWSLT 2013*, 2013.

[2] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[3] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *EACL*, Budapest, Hungary, 2003.

[4] M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel, "The KIT English-French Translation systems for IWSLT 2011," in *Proceedings of the Eight International Workshop on Spoken Language Translation (IWSLT)*, 2011.

[5] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *International Conference on Spoken Language Processing*, Denver, Colorado, USA, 2002.

[6] K. Heafield, "KenLM: faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, United Kingdom, July 2011, pp. 187–197.

[7] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[8] J. Niehues and S. Vogel, "Discriminative Word Alignment via Alignment Matrix Modeling." in *Proceedings of Third ACL Workshop on Statistical Machine Translation*, Columbus, USA, 2008.

[9] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin,

and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of ACL 2007, Demonstration Session*, Prague, Czech Republic, 2007.

[10] M. Mediani, J. Niehues, and A. Waibel, "Parallel Phrase Scoring for Extra-large Corpora," in *The Prague Bulletin of Mathematical Linguistics*, no. 98, 2012, pp. 87–98.

[11] G. F. Foster, R. Kuhn, and H. Johnson, "Phrasetable smoothing for statistical machine translation," in *EMNLP*, 2006, pp. 53–61.

[12] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Sixth Workshop on Statistical Machine Translation (WMT 2011)*, Edinburgh, UK, 2011.

[13] F. J. Och, "An Efficient Method for Determining Bilingual Word Classes." in *EACL'99*, 1999.

[14] E. Cho, J. Niehues, and A. Waibel, "Segmentation and Punctuation Prediction in Speech Language Translation using a Monolingual Translation System," in *Proceedings of the Ninth International Workshop on Spoken Language Translation (IWSLT)*, 2012.

[15] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Pittsburgh, PA, USA, 2005.

[16] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *International Conference on New Methods in Language Processing*, Manchester, United Kingdom, 1994.

[17] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Skövde, Sweden, 2007.

[18] J. Niehues and M. Kolss, "A POS-Based Model for Long-Range Reorderings in SMT," in *Fourth Workshop on Statistical Machine Translation (WMT 2009)*, Athens, Greece, 2009.

[19] T. Herrmann, J. Niehues, and A. Waibel, "Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation," in *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*. Altanta, Georgia, USA: Association for Computational Linguistics, June 2013.

[20] A. N. Rafferty and C. D. Manning, "Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines," in *Proceedings of the Workshop on Parsing German*, 2008.

[21] D. Klein and C. D. Manning, "Accurate Unlexicalized Parsing," in *Proceedings of ACL 2003*, 2003.

[22] J. Niehues and A. Waibel, "Detailed Analysis of Different Strategies for Phrase Table Adaptation in SMT," in *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*, 2012.

[23] R. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the ACL 2010 Conference Short Papers*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 220–224.

[24] A. Mauser, S. Hasan, and H. Ney, "Extending Statistical Machine Translation with Discriminative and Trigger-based Lexicon Models," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1*, ser. EMNLP '09, Singapore, 2009.

[25] J. Niehues and A. Waibel, "An MT Error-Driven Discriminative Word Lexicon using Sentence Structure Features," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, 2013, pp. 512–520.

[26] H.-S. Le, A. Allauzen, and F. Yvon, "Continuous Space Translation Models with Neural Networks," in *Proceedings of the 2012 Conference of the NAACL-HLT*, Montréal, Canada, June 2012.

[27] J. Niehues and A. Waibel, "Continuous Space Language Models using Restricted Boltzmann Machines," in *Proceedings of the Ninth International Workshop on Spoken Language Translation (IWSLT)*, 2012.

[28] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.

[29] D. Varga, L. Németh, P. Halácsy, A. Kornai, V. Trón, and V. Nagy, "Parallel corpora for medium density languages," in *Recent Advances in Natural Language Processing (RANLP 2005)*, 2005, pp. 590–596.

[30] M. Diab, "Second Generation Tools (AMIRA 2.0): Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking," in *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April 2009.

The CASIA Machine Translation System for IWSLT 2013

Xingyuan Peng, Xiaoyin Fu, Wei Wei, Zhenbiao Chen, Wei Chen, Bo Xu

Interactive Digital Media Technology Research Center, Institute of Automation,
Chinese Academy of Sciences, Beijing, 100190, China

{xingyuan.peng, xiaoyin.fu, wei.wei.media, zhenbiao.chen, xubo}@ia.ac.cn

Abstract

In this paper, we describe the CASIA statistical machine translation (SMT) system for the IWSLT2013 Evaluation Campaign. We participated in the Chinese-English and English-Chinese translation tasks. For both of these tasks, we used a hierarchical phrase-based (HPB) decoder and made it as our baseline translation system. A number of techniques were proposed to deal with these translation tasks, including parallel sentence extraction, pre-processing, translation model (TM) optimization, language model (LM) interpolation, turning, and post-processing. With these techniques, the translation results were significantly improved compared with that of the baseline system.

1. Introduction

This paper describes the machine translation (MT) system developed by the Institute of Automation Chinese Academy of Sciences (CASIA) for the evaluation campaign of IWSLT 2013. We participated in the optional MT track with the Chinese-English and English-Chinese translation tasks. Our translation system is based on the hierarchical phrase-based translation model [1]. We used the state-of-the-art HPB translation system as our baseline system.

Efforts have been made to improve the translation performance. To obtain high quality parallel sentences, we introduced a parallel sentence extraction method based on the lexical translation probabilities. Rule-based translation of named entities was proposed to deal with translations of time and numbers that are done incorrectly. For our translation model training, the forced alignment technique [2] was used for optimizing the translation rules and reducing the hierarchical phrase table size. In addition, we used the provided monolingual corpora to train different language models and interpolated the language model to adapt to the translation tasks. At last, we added word-to-word phrases to the hierarchical phrase table to reduce the number of untranslated words.

The remainder of the paper is structured as follows. Section 2 describes the resources used in our system. Section 3 gives an overview of the whole system. Section 4 discusses the improvement of translation performance in detail. Section 5 presents the experiments and evaluation results. Finally, section 6 concludes the paper.

2. Resources Used in IWSLT 2013

Training of the translation models and language models for MT systems is constrained to data supplied by the organizers. Since we only participated in the Chinese-English and English-Chinese translation tasks, we made full use of the parallel and monolingual training corpora in Chinese and English. The training corpora are divided into two parts: the parallel data for the TM training and the monolingual data for the LM training.

Parallel Data: There are two types of parallel data for our translation model training.

- WIT^3 (Web Inventory of Transcribed and Translated Talk), which redistributes the original content published by the TED Conference website [3]. After the pre-processing, there are 151,149 aligned English-Chinese parallel sentences in these data. We will use WIT^3 to represent the extracted parallel sentences from the WIT^3 training corpus.
- MultiUN (Multilingual UN Parallel Text 2000-2009), which provides parallel corpus extracted from the United Nations website [4]. As the data has alignment problems, we realigned the parallel sentences (Details will be introduced in section 4.1). Finally, 7,819,291 parallel sentences were extracted in these data. We will use UN to represent the parallel sentences extracted from the MultiUN training corpus.

Monolingual Data (English): There are five English datasets used for the LM training in our experiment.

- News Commentary v7 from WMT 2012
- News Crawl from WMT 2012
- Europarl v7
- LDC2011T07 English Gigaword Fifth Edition
- Monolingual UN corpus (the English part of MultiUN)

Monolingual Data (Chinese): There are three Chinese datasets used for the LM training in our experiment.

- WIT^3 (the Chinese part of the parallel data mentioned above).

- Monolingual UN corpus (the Chinese part of MultiUN)
- Google book grams

3. System Overview

3.1. Chinese Word Segmentation System

The Chinese word segmentation (CWS) system is based on our in-house toolkit [5], which combines both CRF-based model and N-gram language model to segment Chinese words. CRF model treats the CWS task as a sequence tagging question. It overcomes the tagging bias problem in generative models. However, it tends to generate longer words, which is harmful to SMT system because it causes data sparseness. To overcome this drawback, we introduced N-gram language model as a supplement to CRF-based model. The N-gram language model generates significantly shorter words than the CRF-based model does, which can be helpful to distinguishing shorter words. Compared to the open source CWS toolkit ICTCLAS[1], the CRF++[2] training toolkit was used to train our CRF based model and the SRILM toolkit was used to train the N-gram language model with the annotated Chinese People's Daily News corpus as resources from February to June, 1998. We tested the performance on the news corpus in January, 1998. The results measured by precision (**P**), recall (**R**) and **F1** measure are listed in Table 1.

Table 1: *The CWS results on the Chinese People's News corpus.*

System	**P**	**R**	**F1**
ICTCLAS	98.1%	98.7%	98.4%
CASIA	97.5%	97.7%	97.6%

3.2. Hierarchical Phrase-based System

For our HPB translation system, we employed an in-house implementation of the state-of-the-art MT decoder, which is mainly based on the work of [7]. In HPB translation system, a weighted synchronous context-free grammar is induced. There are two types of phrases distinguished by the non-terminals in HPB rules. Phrases without non-terminals are the initial phrases and those with up to two non-terminals are the hierarchical ones. Both of them were heuristically extracted from the aligned parallel sentences. The search was carried out on a CKY parser with beam search together with a post-processor for mapping source language derivations to target ones. The standard features integrated into our decoder include: phrase translation probabilities and lexical probabilities in both translation directions, word and phrase penalty, glue rules, and N-gram language model, all of which are assigned by the log-linear model [8]. Besides, we used the cube pruning [9] to speed up our decoder, and the standard MERT [10] to tune the weights of our features on the 100-best translation assumptions on IWSLT 2010 development set.

[1] http://www.ictclas.org/

[2] http://crfpp.sourceforge.net/

3.3. Forced Alignment System

Usually, the original HPB phrases can be extracted heuristically from the aligned words of parallel sentences, as proposed in [7]. However, the heuristical phrases extraction suffers from a large amount of redundant rules and meets difficulties in probability estimation. To avoid these, we employed the idea of force-aligning training data with the heuristically trained HPB rules [2]. Instead of directly applying these HPB rules in decoding, we used the original HPB rule to align the parallel training sentences and generated the bilingual derivation trees that represent both the source and target sentences. Then, HPB rules were extracted from the derivation trees with a threshold pruning. The translation probabilities of HPB rules were updated.

It should be noted that we only re-estimated the phrasal translation probabilities, and kept the lexical translation probabilities estimated with the method of [11]. After generating the optimized HPB rules, we tested our forced alignment (FA) method on the IWSLT 2012 and 2013 Chinese-English MT test set with the translation models trained from the WIT^3 parallel corpus. The phrase table sizes and translation results are listed in Table 2.

Table 2: *Forced Alignment results on the IWSLT 2012 and 2013 Chinese-English translation tasks. The translation performances are measured by BLEU and TER.*

System	tst2012		tst2013		#Phrases
	BLEU	TER	BLEU	TER	
WIT^3	12.5	67.0	14.3	68.4	22.7M
WIT^3+FA	12.6	66.7	14.4	67.7	11.4M

The result showed that the total HPB phrase table was reduced by 50% and the performances in both translation tasks are slightly better compared to that of the baseline HPB translation system. Besides, a large number of phrases have been dropped out by our forced alignment, speeding up the HPB decoder.

4. Improvements

4.1. Parallel Sentence Extraction

The MultiUN Chinese-English parallel corpus provided by the IWSLT2013 Evaluation Campaign is aligned by chapter instead of sentence. It is difficult to train word alignment using this corpus. By investigating the MultiUN dataset, we found two alignment problems. First, instead of one to one sentence alignment, the sentence on the source side may

align to two or even more sentences on the target side. Second, the sentence on the source side may have no aligned sentences on the target side. The simple introduction of the MultiUN corpus may not help to improve the translation performance. Therefore, we proposed a method to extract parallel sentences from the MutiUN dataset.

Given the source sentence f with m words and target sentence e with n words, we suppose that words on the source side can be aligned to any words on the target side. The similarity between the two sentences is calculated as

$$\begin{aligned} sim(f,e) &= \lambda_1 P(f|e) + \lambda_2 P(e|f) \\ &+\lambda_3 L(e) + \lambda_4 L(f) + \lambda_5 R(f,e) \end{aligned} \quad (1)$$

where $P(f|e)$ is the average weights of the words in target sentences that are aligned to those in source sentences. It can be calculated as

$$P(f|e) = \frac{1}{m}\sum_{i=1}^{m}(\frac{1}{n}\sum_{j=1}^{n} log(p_{ij})) \quad (2)$$

where p_{ij} is the lexical translation probability and i and j are the position of words in source and target sentences. If there is no lexical translation probability between the aligned words, we set p_{ij} to be a minimal probability with e^{-10}. $P(e|f)$ can be calculated in the similar way.

$L(f)$ and $L(e)$ are used to punish the sentence length, which can be calculated as:

$$L(f) = log(m) \quad (3)$$

$$L(e) = log(n) \quad (4)$$

$R(f,e)$ is used to punish the length difference between the aligned sentences:

$$R(f,e) = log(max\{m,n\}/min\{m,n\}) \quad (5)$$

In our experiment, we supposed that the source sentence could align to at most 10 target sentences. All the possible alignments were scored by Equation 1, and the aligned sentences with the highest score were selected as the parallel sentence pairs. For the MultiUN parallel data, we finally obtained 7,819K sentence pairs to train our translation model.

4.2. Rule-based Translation of Named Entities

Although some named entities as time and numbers can be well translated by translation models, a majority of them cannot be correctly translated. Therefore, we introduced rule-based translation of named entities toolkit, which identifies time and number entities from the source sentences, and then translates it into the target language. We did not treat the named entities in the post-processing, but introduced them as normal translation rules. In our translation tasks, we first built a phrase table containing the named entities along with the translation results. Then we added it to the hierarchical translation model with a higher probability during the decoding process.

Take the phrase "26.5 million" in English-Chinese translation tasks as an example. Our toolkit will give us a parallel phrase as "26.5 million ||| 2,650 万". By adding it to the translation model, the named entity of "26.5 million" can be translated correctly.

4.3. Translation Model Optimization

In TM training steps, we used the open source toolkit Giza++ [12] to get the bidirectional word alignments and combined them with $grow\text{-}diag\text{-}final\text{-}and$ method. Then we extracted the initial phrases and hierarchical phrases with heuristic extraction method to generated our original HPB model. We did forced alignment with the original HPB model on our training data and re-estimated the translation probabilities with the extracted phrases. At last, we used these refined phrases to generate our TM model for translation.

4.4. Language Model Interpolation

The language models used in our system are obtained by interpolating individual language models trained on the corpora of a different domain.

For the English language model, these training data sources are mentioned in section 2. First, the 5-gram modified Kneser-Ney discounted LMs are trained by using the SRILM toolkit [6]. Then the optimal interpolation weights for each LMs are estimated by using the tst2011 as the perplexity calculation text. The perplexities of each individual LMs and the final English LM are shown in table 3.

Table 3: *Perplexity and interpolation weights of the 5-gram English Language Models.*

data	tst2011	tst2013	weight
News Com	145.9	143.5	0.127
News Cra	88.8	93.0	0.697
Europarl	291.7	271.0	0.065
LDC Gigaword	403.7	345.7	0.109
UN	114.8	108.9	0.002
interpolate	84.3	84.1	-
prune	103.8	90.1	-

For the Chinese language model, four 4-gram modified Kneser-Ney discounted LMs are trained firstly. Then the optimal interpolation weights are estimated by using the tst2011. During weight estimating, the tst2010-2012 set dose not include the tst2011. However, during interpolating, the tst2010-2012 set includes dev2010, tst2010, tst2011 and tst2012, making the final Chinese LM contain the data of tst2011. Table 4 presents the perplexities of each LM.

4.5. Translated Rule Addition

In our HPB translation system, some of the words in source language are untranslated as no matched rules are available. However, these words actually have translations which can-

Table 4: *Perplexity and interpolation weights of the 4-gram Chinese Language Models.*

data	tst2011	tst2013	weight
WIT^3	188.9	217.5	0.630
UN	575.5	595.5	0.119
Google book grams	4553.1	4444.5	0.110
tst2010-2012	48.8	395.9	0.141
interpolate	83.4	205.2	-

not be extracted because of the restriction during phrase extraction. To avoid the non-translated phenomenon, we extracted word-to-word translation rules for these untranslated words from the lexical translations from word alignment.

For each untranslated word w_f in the source language, we looked up all of its target word w_e from the lexical probability table. The joint probability for each word pair is scored as:

$$P(w_f, w_e) = \frac{1}{2}(logP(w_f|w_e) + logP(w_e|w_f)) \quad (6)$$

where $P(w_f|w_e)$ and $P(w_e|w_f)$ are the bidirectional lexical probabilities.

We chose 3-best joint probabilities with the corresponding translations and added them to the hierarchical phrase table by a very lower probability. With the help of these additional phrase rules, some of the untranslated words could be translated correctly.

5. Experimental Results

We first trained our baseline HPB system (WIT^3) using the extracted WIT^3 parallel corpus. Then we did forced alignment with the baseline HPB model on the WIT^3 training data and obtained the optimized translation model (WIT^3+FA). We added the extracted parallel sentences from UN to WIT^3 and trained a larger translation model (WIT^3+UN). Considering the huge amount of parallel sentences in UN, we copied the WIT^3 corpus five times when combining these two types of parallel sentences. We also used the new translation model to do forced alignment on WIT^3 (WIT^3+UN+FA), which helps to generate more useful translation rules than the smaller translation model. At last, we added the translation rules, which were generated by the named entities toolkit and untranslated words, to our final HPB model as the translated template (WIT^3+UN+FA+Template). Both of Chinese-English and English-Chinese translation models are trained following the same way as described above. The results for Chinese-English and English-Chinese translation tasks on IWSLT 2012 and 2013 test sets are listed in Tables 5 and 6.

In Tables 5 and 6, the first two systems are trained using only the WIT^3 corpus. By adding the UN corpus to WIT^3 corpus, the translation performance was improved on both of the Chinese-English translation tasks, indicating that our extraction method can effectively get parallel sentences from MultiUN training corpus. However, the improvement on English-Chinese translation tasks is not significant as that on Chinese-English tasks. The results also show that our forced alignment can get better performance on the tasks with much smaller translation models. Moreover, the introduction of translation rules for named entities and untranslated words gives us the best results on the IWSLT 2013 translation tasks.

It is noteworthy that the satisfactory English-Chinese translation results on tst2012 are not attributed to our high quality translation system, but the incorrectly trained LM interpolated with data from tst2012.

Table 5: *Results for the Chinese-English MT task on IWSLT 2012 and 2013 test sets. The primary submission is the system combination of all the training methods.*

System	tst2012		tst2013	
	BLEU	TER	BLEU	TER
WIT^3	12.5	67.0	14.3	68.4
WIT^3+FA	12.6	66.7	14.4	67.7
WIT^3+UN	12.8	67.4	14.7	68.1
WIT^3+UN+FA	12.9	66.0	14.8	66.8
WIT^3+UN+FA+Template	13.0	65.8	15.0	66.4

Table 6: *Results for the English-Chinese MT task on IWSLT 2012 and 2013 test sets. The primary submission is the system combination of all the training methods.*

System	tst2012		tst2013	
	BLEU	TER	BLEU	TER
WIT^3	12.7	71.2	11.9	70.1
WIT^3+FA	12.8	70.6	11.9	69.6
WIT^3+UN	12.9	71.3	12.1	69.9
WIT^3+UN+FA	12.9	70.9	12.2	69.7
WIT^3+UN+FA+Template	13.0	70.7	12.3	69.6

6. Conclusion

In this paper, we presented our submission runs to the IWSLT 2013 Evaluation Campaign for the optional MT track on Chinese-English in both directions. We did our translation tasks by using the in-house hierarchical phrase-based decoder and Chinese word segmentation system as well as other open source toolkits. In particular, we used the forced alignment to optimize the HPB rules, which obtained the same translation results with a much smaller phrase table. To get better translation performances, we introduced the template rules into our decoder to deal with time and number entities and the words that exist in training data but do not have translation rules.

In future work, we plan to add some other features to our log-linear model and use the system combination methods to modify our system.

7. Acknowledgements

This work was supported by the National High Technology Research and Development Program (863 Program) of China under Grant No.2011AA01A207.

8. References

[1] D. Chiang, "A hierarchical phrase-based model for statistical machine translation", in Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, 2005, pp.263–270.

[2] X. Fu, W. Wei, S. Lu, B. Xu, "Filtration and Optimization for Hierarchical Phrase-based Model with Forced Alignment", in Proceedings of the 12th China National Conference on Computational Linguistics (CCL), 2013.

[3] M. Cettolo, C. Girardi and M. Federico, "WIT3: Web Inventory of Transcribed and Translated Talks", In Proceedings of EAMT, 2012, pp.261–268.

[4] A. Eisele and Y. Chen, "MultiUN: A Multilingual corpus from United Nation Documents", in Proceedings of LREC, 2010.

[5] W. Chen, W. Wei, Z. Chen and B. Xu, "Integrating Multi-source Bilingual Information for Chinese Word Segmentation in Statistical Machine Translation", in Proceedings of the 12th China National Conference on Computational Linguistics (CCL), 2013.

[6] A. Stolcke, "SRILM: An Extensible Language Modeling Toolkit", in Proceedings of the 7th International Conference on Spoken Language Processing, 2002, pp.901–904.

[7] D. Chiang, "Hierarchical phrase-based translation" Computational Linguistics. 33(2), 2007, pp.201–228.

[8] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation", in Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp.295–302.

[9] L. Huang and D. Chiang, "Forest Rescoring: Faster Decoding with Integrated Language Models", in Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, 2007, pp.144–151.

[10] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation", in Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics, 2003, pp.160–167.

[11] P. Koehn, F. J. Och, D. Mareu, "Statistical Phrase-Based Translation", in Proceedings of the 2003 Conference of the NAACL, 2003, pp.48–54.

[12] F. J. Och and H. Ney, "A Systematic Comparison of Var-ious Statistical Alignment Models", Computational Linguistics , 29(1), 2003, pp.19–51.

Using Viseme Recognition to Improve a Sign Language Translation System

Christoph Schmidt, Oscar Koller, Hermann Ney [1]
Thomas Hoyoux, Justus Piater [2]

[1]Human Language Technology and Pattern Recognition Group,
RWTH Aachen University, Aachen, Germany
{surname}@cs.rwth-aachen.de

[2]Intelligent and Interactive Systems,
University of Innsbruck, Austria
{firstname}.{surname}@uibk.ac.at

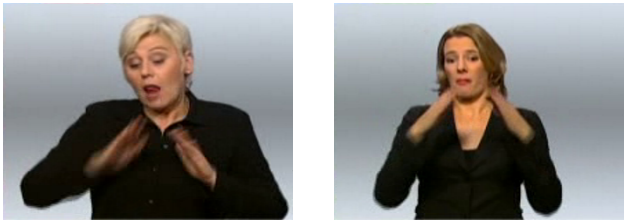
Abstract

Sign language-to-text translation systems are similar to spoken language translation systems in that they consist of a recognition phase and a translation phase. First, the video of a person signing is transformed into a transcription of the signs, which is then translated into the text of a spoken language. One distinctive feature of sign languages is their multi-modal nature, as they can express meaning simultaneously via hand movements, body posture and facial expressions. In some sign languages, certain signs are accompanied by mouthings, i.e. the person silently pronounces the word while signing. In this work, we closely integrate a recognition and translation framework by adding a viseme recognizer ("lip reading system") based on an active appearance model and by optimizing the recognition system to improve the translation output. The system outperforms the standard approach of separate recognition and translation.

1. Introduction

The aim of a sign language-to-text translation system is to translate a video of a person signing into a text in a spoken language. Similar to spoken language translation systems, such a system consists of a recognition component in which the individual signs are recognized, and a translation component in which the sequence of signs is translated into a text of the spoken language. The translation step is necessary as signed languages, if evolved naturally, differ at great length from spoken languages, having a unique grammar and vocabulary.

Sign languages are multi-modal in the sense that they express meaning simultaneously via different communication channels. Besides the manual information such as hand shape, orientation and movements, non-manual aspects such as body posture and facial expressions play a vital role in expressing meaning. In countries which have a strong oral education tradition, e.g. Germany, some signs are accompanied by mouthings, i.e. the signer pronounces the spoken language word with his lips while signing with his hands. These mouthings are particularly used to derive new signs by using the hand movements of a similar or more general sign and changing only the mouthing. In the example in Figure 1, the particular sign for the Alps is derived by depicting the form of a mountain while silently pronouncing the word Alps (German: "Alpen").



ALPS (mouthing "Alpen") MOUNTAIN (mouthing "Berg")

Figure 1: Two signs with the same manual component, differing only in the mouthing. At the time of the snapshots, the underlined letters are pronounced.

In this work, we want to use a mouthing recognition system, which is often also referred to as a visual speech recognition system, to improve the quality of the translation system by providing the mouthing as an additional input to the translation system and by exploiting the correspondence between the mouthings and spoken language words. Moreover, we achieve a close integration of recognition and translation by optimizing the recognition system with respect to the translation output. The approach is depicted in Figure 2.

This paper is structured as follows: First, we present related work in Section 2. The RWTH-Phoenix-Weather corpus, which is used in our experiments, is described in Section 3. In Section 4, we outline the technique of active appearance models, which we apply to track the face and the mouth region. The mouth shape and opening is then used to recognize viseme sequences in Section 5. We present our experimental results in Section 6. Conclusions and an outlook are given in

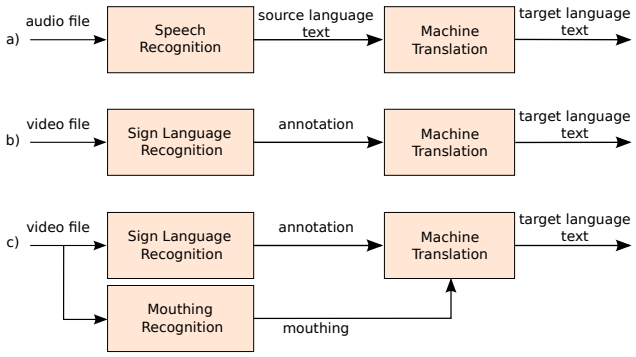


Figure 2: System architectures of a) spoken language translation, b) sign language translation and c) the system proposed in this work including a mouthing recognition

Section 7.

2. Related Work

To track the position of the face and the mouth, we apply an active appearance model. From the resulting locations of the mouth corners, we calculate high-level features such as the degree of opening, which we use to train a viseme recognition system.

[1] and [2] use active appearance models to recognize a predefined set of facial expressions. [3] and [4] provide facial features for the use in a sign language recognition framework, i.e. they integrate low-level facial features into their system to improve the recognition of the signs.

There are several approaches to viseme recognition. We follow the geometric approach, using distances between lips, chin and nose to train a recognition system. This approach is similar to [5], who use active contours ("snakes") to detect the lips. Their approach is more sophisticated, as they calculate histograms of the area inside the lips to detect tongues and teeth, while our approach is more general in that the active appearance models which are trained on the whole face also detect other facial features such as eyebrow raise and cheek movements.

Sign language machine translation faces the challenge that corpus resources are particularly sparse. A thorough overview of sign language translation is given in [6].

3. The RWTH-Phoenix-Weather Corpus

The RWTH-Phoenix-Weather corpus is a video-based, large vocabulary corpus of German Sign Language recorded and annotated for the use in statistical pattern recognition and statistical machine translation. The public TV broadcasting station Phoenix regularly broadcasts the major public news programs with an additional interpretation into German Sign Language using an overlay window which shows the interpreter.

The RWTH-Phoenix-Weather corpus contains the weather forecast portions of these news programs, which were manually annotated by a deaf expert and revised by a hard-of-hearing expert. The weather forecasts were chosen because weather forecasting forms a rather compact domain with a limited vocabulary. A complex domain such as news programs would require a much larger corpus to reliably estimate statistical models, but annotating such a corpus was infeasible due to time and budget constraints.

Since sign language expresses meaning simultaneously via hand movements, body posture, facial expressions, mouthing, etc., one open question in the sign language research community is how to capture this multi-modal nature of sign languages in a comprehensive annotation system. A simple annotation method is gloss annotation, where a sign is annotated by one or several words which roughly correspond to its meaning, usually written in the stem form in upper case. Since the same sign can have several meanings in different contexts, it can be transcribed differently depending on its context. In contrast to this, the term ID-gloss [7] is used if one sign is always annotated with the same gloss, independent of its meaning in a particular context. In our corpus and experiments, we use ID-glosses.

The annotation of a sign language video corpus highly depends on the task at hand. For example, if a linguist wants to study certain linguistic patterns, the annotation should be detailed with respect to these patterns. In the same way, an annotation suitable for an automatic sign language recognition system should be tailored according to the features which the system can actually recognize. Since the RWTH-Phoenix-Weather corpus was originally developed for the recognition of hand-based features, both the time boundaries of the ID-glosses and their label were mainly based on the signing hands. This means that signs which are identical in the hand components but differ in their mouthing received the same label. For example, the sign of a specific mountain is formed by mouthing its name and performing the general sign for mountain with the hands (see Figure 1). In the corpus, both variants were glossed as "MOUNTAIN", because they could not be distinguished by the hand features used at the time.

In addition to the annotation of the ID-glosses, the RWTH-Phoenix-Weather corpus has been marked with time boundaries on the sentence as well as the gloss level. The spoken German weather forecast has been transcribed semi-automatically using a state-of-the-art automatic speech recognition system. To train active appearance models on this corpus, facial landmarks have been manually labeled on a small set of images.

In the following, we will briefly describe the corpus setup and statistics. For a more thorough description see [8]. Note that in this setup, we use only the portions of RWTH-Phoenix-Weather for which time boundaries for individual glosses are annotated. These are necessary to extract the features for the viseme recognizer.

	DGS	German
# signers	7	
# editions	190	
duration[h]	3.25	
# frames	293,077	
# sentences	2,552	
# running glosses	14,771	30,860
vocabulary size	911	1,452
# singletons	120	337

Table 1: Statistics of the RWTH-Phoenix-Weather corpus for DGS and announcements in spoken German

Figure 3: Visualization of facial annotations

The corpus statistics for the RWTH-Phoenix-Weather corpus with time boundaries for individual glosses can be found in Table 1. The database features a total of seven interpreters, consists of 2640 sentences and a total of 14,771 running glosses. Baseline translation results both from German to German Sign Language and in the opposite direction can be found in [9]. Sign language corpora are much smaller than spoken language corpora for two reasons. Since there is no standard writing system for sign languages, sign language corpora containing a written notation do not exist by themselves but have to be produced by experts who define a suitable annotation scheme for the task at hand. Moreover, annotating a video corpus is quite time consuming, because the annotators have to mark time boundaries of individual signs and have to use a canonical notation for sign variants which are frequent.

To train active appearance models on this corpus, 38 facial landmarks for all seven interpreters have been labeled in a total of 369 images (that is, about 50 images per interpreter). Care was taken in selecting a set of images which contain many different expressions, including extreme ones, such that the trained models can approximately represent a large span of expressions for each interpreter. Two examples of the facial annotations are shown in Figure 3.

4. Active Appearance Models

The facial features which are used for recognizing the signer's mouthing consist of continuous measurements of some quantities related to mouthing, such as horizontal and vertical mouth openness, and other facial cues such as eye brow raise. As shown in Table 2, these measurements are based on lower-level facial features which are defined as a set of consistent, salient point locations on the interpreter's face. As illustrated in Figure 3, these fiducial points – also called landmarks – correspond to key locations on the cheeks and chin outlines, the nose ridge and nose base, the eyelids and eye corners, the eyebrow outlines and the lip and mouth corners. We wish to track those point features accurately in the sign language videos in order to extract the higher-level facial features which will in turn be used to recognizing the words pronounced by the signer. Since the structure of the human face as described by a set of such point features exhibits a lot of variability due to changes in pose and expression, we chose to base our tracking strategy on the deformable model registration method known as active appearance models.

Semantic description	Related point features #
mouth vertical openness	{18, 21, 24, 25, 26, 27}
mouth horizontal openness	{18, 21}
lower lip to chin distance	{26, 27, 32, 33}
upper lip to nose distance	{15, 16, 17, 18, 21, 24, 25}
left eyebrow state	{0, 1, 2, 6, 8}
right eyebrow state	{3, 4, 5, 10, 12}
gap between eyebrows	{2, 3}

Table 2: High-level facial features used in the proposed clustering approach and the related lower-level point features (Figure 3)

Active appearance models (AAMs), first proposed in [10] and notably reformulated in [11], are a popular instance of the family of deformable model methods for image interpretation. Such model-based methods attempt to recover an object's structure as it appears in an image by registering a deformable shape model of the object to the image data. Mathematically, the shape $\mathbf{s}$ of an object is defined as the vector of stacked coordinates of its v landmark points:

$$\mathbf{s} = (x_1, y_1, x_2, y_2, \ldots, x_v, y_v)^\intercal$$

assuming here that each landmark is a 2-dimensional point representing a semantically meaningful part of the object, such as an eye corner in the human face.

AAMs model shape deformation using a so-called point density model (PDM), which is a parametric linear subspace model learned statistically by principal component analysis (PCA) on a set of training shape examples. These examples are given as expert annotations of images of the object of interest, such as shown in Figure 3 for the human face. In such a representation, any shape $\mathbf{s}$ of the deformable object can be expressed by the generative model as a base shape $\mathbf{s}_0$ plus a linear combination of n shape vectors $\mathbf{s}_i$:

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{n} p_i \mathbf{s}_i$$

Registering a PDM to the image data then reduces to finding the optimal coefficient values p_i of this linear combination,

i.e. the optimal PDM's parameters. AAMs propose to model the coupling between the PDM and the image data, i.e. the predictions on the PDM's landmarks locations given a target image, via a holistic appearance model of the pixel intensity values of the object's image. This appearance model is again a parametric linear subspace model, obtained by applying PCA to shape-normalized training example images of the object of interest. This shape normalization involves the warping of every example image to a reference frame, which is typically done by piecewise affine warping functions defined between each example shape and the base shape $\mathbf{s}_0$ of the PDM. The generative appearance model is then used to express any object's appearance $A(\mathbf{x})$ as a base appearance $A_0(\mathbf{x})$ plus a linear combination of m appearance images $A_i(\mathbf{x})$:

$$A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) \qquad \forall \mathbf{x} \in \mathcal{R}(\mathbf{s}_0)$$

where $\mathcal{R}(\mathbf{s}_0)$ denotes the set of pixel locations within the region defined by the base shape $\mathbf{s}_0$, i.e. the reference frame for the object's appearance.

Given these two generative models and following the so-called "independent" AAMs formulation proposed in [11], registration can be seen as an image matching problem between the synthetic model image and the shape-normalized target image; the fitting goal can therefore be expressed as finding the parameters $\mathbf{p} = (p_1, p_2, \ldots, p_n)^\intercal$ and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_m)^\intercal$ that minimize the following sum of squared differences:

$$\sum_{\mathbf{x} \in \mathcal{R}(\mathbf{s}_0)} \left[A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \right]^2$$

where I is the target image and $\mathbf{W}(\mathbf{x}; \mathbf{p})$ is a (piecewise affine) warping function which projects a pixel location $\mathbf{x}$ from the reference frame to the target image frame, depending on the PDM's parameters $\mathbf{p}$. The minimization of this quantity is non-linear in the parameters $\mathbf{p}$ and must be solved iteratively by linear approximation, typically using the Gauss-Newton algorithm.

Variants met in the AAM-related literature mostly differ in the way they parameterize this linear approximation to derive the parameters update equation. In this work, we chose to use the efficient version of the simultaneous inverse-compositional AAM (SICAAM) proposed in [12]. This variant is more robust than others to large variations in shape and appearance, which typically occur when dealing with facial expressions in the context of sign language. Moreover, in order to cope with large off-plane head rotations, which are also common in sign language and can lead a 2D AAM to failure, we used the refinement proposed in [13]. In this work, a 3D PDM is estimated using a non-rigid structure-from-motion algorithm on the training shapes, and is then involved in the optimization process which incorporates a regularization term encouraging the 2D shape controlled by the 2D PDM to be a valid projection of the 3D PDM. Similar to the 2D PDM, the 3D PDM expresses any 3D shape $\mathbf{S}$ as a 3D base shape $\mathbf{S}_0$ plus a linear combination of $\bar{n}$ 3D shape vectors $\mathbf{S}_i$:

$$\mathbf{S} = \mathbf{S}_0 + \sum_{i=1}^{\bar{n}} \bar{p}_i \mathbf{S}_i$$

Notice that the 3D PDM is also involved in the calculation of the high-level facial features described below.

The procedure for the production of the high-level facial features includes a training stage:

1. Extrude the set of 2D training shape examples to 3D by means of the 3D PDM.
2. Remove global translations and rotations by aligning every extruded shape to the base shape $\mathbf{S}_0$ of the 3D PDM.
3. Project the aligned extruded shapes to 2D and, for each, estimate local area-based measurements corresponding to the point features subsets given in Table 2.
4. For each point features subset, store as the training output the minimum and maximum values of the corresponding local area-based measurements.

Extracting high-level facial features from the tracked lower-level point features is then done in the following way:

1. Extrude the registered shape and remove its global translation and rotation by means of the 3D PDM
2. Project the aligned extruded shape to 2D and, for each point features subset given in Table 2, estimate the corresponding local area-based measurement.
3. Normalize each local area-based measurement between 0 and 1 according to the minimum and maximum values obtained during training for the corresponding point features subset.
4. Each registered shape is then associated with a vector of D (in our work $D = 7$) continuous values in the range $[0, 1]$, corresponding to our high-level facial features.

Seven SICAAMs specific to the seven interpreters of RWTH-Phoenix-Weather have been trained for the end purpose of extracting high-level facial features from the gloss-annotated videos as shown in Figure 4. Training and tracking with one single SICAAM for all seven interpreters would have been a viable choice as well because of the enhanced robustness of this AAM variant to variability in identity. However, we wanted to obtain the best possible accuracy in the tracking of the low-level point features. On the other hand, the calculation of our high-level features is rather sensitive to identity changes and as such had to be designed in an identity-dependent fashion. The extraction of reliable

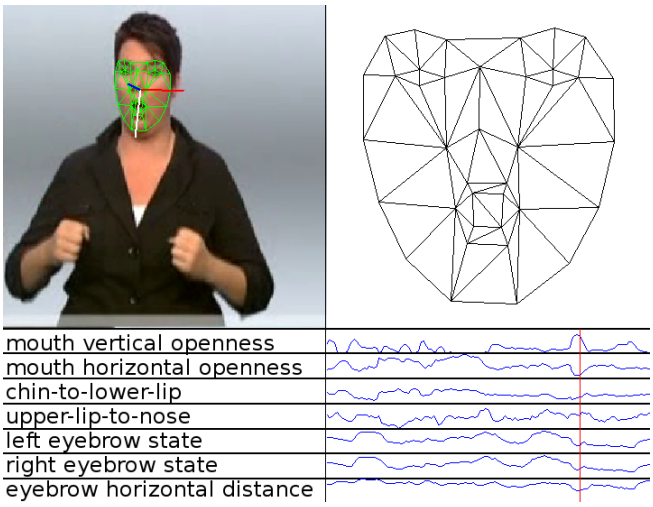


Figure 4: High-level feature extraction
Top left: the grid of fitted AAM points
Top right: rotated and normalized AAM points
Bottom: high-level feature values over time

identity-*independent* facial features similar to those used in this work is part of the advanced computer vision research topic known as "expression transfer" and is beyond the scope of this paper, where our primary goal is to give a proof of concept that including the mouthing information from a viseme-based mouthing recognizer can improve a sign language translation system. The mouthing recognizer will be described in more detail in the next section.

5. Viseme Recognition

Since the RWTH-Phoenix-Weather corpus was mainly annotated for the use in sign language recognition of hand-based features, mouthings have not been annotated for the whole corpus. To obtain possible candidates for the words the signer has pronounced while signing, we align the glosses denoting the signs with their translation in the spoken language. We use the open-source toolkit GIZA++ to align each gloss to at most one word. However, not all signs are accompanied by mouthing. We therefore include a silence model representing no mouth movement and a garbage model for mouthing gestures not representing specific viseme sequences in the viseme recognizer. To train a viseme recognizer on the videos, we need a viseme transcription of the spoken words. We first use a lexicon from our speech recognition system trained on German to lookup each German word which is aligned to a gloss and to find its corresponding sequence of phonemes. As many phonemes cannot be visually distinguished, for example the phonemes P and B differ only in the aspiration which is not visible, we further map the set of phonemes to a set of visemes, i.e. visually distinguishable phonemes. We follow the suggestion of [14] and map the set of phonemes to a set of 15 visemes. A list of the used visemes can be found in Table 3.

Phoneme	Viseme	Examples
p, b	P	Pause, Bitte
t, d, k, g	T	Tonne, Dach, König, Gier
n, @n, l, @l	N	Nadel, raten, Liebe, Igel
m	M	Mutter
f, v	F	Finder, Vase
s, z	S	Fass, Stein
S, Z, tS, dZ	Z	Schein, Garage, Tscheche
h, r, x, N	R	Hase, Reden, Dach, Wange
j, C	C	Junge, Wicht
i:, I, e:, E:, E	E	Bier, Tisch, Weg, Räte, Menge
a:, a	A	Wagen, Watte
o:, O	O	Wolle, Wogen
u:, U	U	Buch, Runde
@, 6	Q	Bitte, Weiher
y:, Y, 2:, 9	Y	Tür, Mütter, Goethe, Götter

Table 3: Phoneme-viseme mapping (taken from [14])

Statistics on the aligned gloss translation pairs allow to exclude noisy alignments. Specifically, this is done by using an empirically set threshold of at least four occurences per gloss translation pair and considering only translation alignments that represent at least 10% of all translations for a specific gloss. Gloss translation alignments which do not meet these requirements are put into the garbage model.

We then train our state-of-the art speech recognition system RASR [15] using 15 viseme hidden Markov models (HMMs) and the garbage model, each containing three states with single Gaussian densities, a globally pooled covariance matrix and global time distortion penalties. Silence visemes are represented by an additional single state HMM. The models are fed with the seven high-level facial features. A modelling lexicon defining possible pronounciation variants for each gloss is provided to the system. It is generated based on the statistics on the aligned gloss translation pairs. The system is initialized with a linear segmentation on the RWTH-Phoenix-Weather data providing gloss time boundaries. The EM-algorithm with viterbi approximation iteratively accumulates the HMMs and uses them to re-estimate the state-frame-alignment, while choosing the most likely pronounciation variants representing different sequences of visemes. This process can be considered as weakly supervised clustering. After 10 iterations the algorithm converges to a stable optimum, yielding the hypothesized viseme sequences for each gloss. In order to remove outliers we chose the RANSAC algorithm [16] to further refine the state-frame-alignment and hence the models.

Table 5 shows the achieved performance of the viseme recognizer after each of its training and refinement steps. The Character Error Rate (CER) compares the hypothesized viseme sequence on the character level to 640 manually annotated mouthings.

Subsequently, the hypothesized viseme sequences are fil-

	CER	Recall
initial segmentation	40.5	82.5
10x EM-realignment	35.7	47.5
after RANSAC processing	32.2	45.5

Table 4: Character Error Rate (CER) and recall in [%] of viseme recognizer measured on 640 manual annotations.

tered by comparing them to the original GIZA++ alignment and estimating the relative error for a given gloss and viseme sequence. Viseme sequences that cause a high mismatch to the GIZA++ alignment are less likely to support the following translation step. We tested different error thresholds on the development set and obtained best results for a threshold of 30. Translation variants with a relative error higher were removed, that is, no gloss variant was generated.

6. Experiments

For our experiments, We use the open-source translation system JANE [17]. The training corpus is word-aligned using GIZA++, and phrase pairs consistent with this alignment are extracted. Previous experiments on this corpus ([9]) have shown that phrase-based systems outperform hierarchical systems, and consequently we choose a phrase-based system for machine translation. Since the corpus is very small, regular MERT training on a held-out development set leads to unstable optimization parameters. We therefore apply a technique similar to cross-validation where we train five different systems, each with a different portion of the training data used as the development set. In each optimization iteration, we concatenate the n-best lists of each individual system and optimize the parameters on this concatenated list.

The baseline system consists of a two-stage approach in which the glosses with no additional information are translated. This corresponds to part b) in Figure 2.

In the approach proposed in this work, which is depicted in part c) of the same figure, we add the mouthing information obtained from the viseme recognizer as an additional knowledge source to the translation system. This is done in the following way. In cases in which the viseme recognizer has a high confidence to recognize a word correctly, we split up the gloss into several variants. E.g., the gloss MOUNTAIN(="BERG") from Figure 1 could be split up into two gloss variants MOUNTAIN_alps and MOUNTAIN_mountain. The machine translation system is then trained on these gloss variants.

Since the mouthing usually corresponds to a word in the spoken language, we want to increase the probability of the gloss variants which are translated into their mouthing component. This can be done on the word and the phrase level.

On the word level, we increase the probability of the IBM1-like lexical smoothing of such pairs by a factor α. The factor is optimized on the development set.

System	Dev		Test	
	BLEU	TER	BLEU	TER
Baseline	35.5	58.8	23.8	66.5
Oracle	36.8	53.4	29.8	60.1
+ word level	39.8	45.3	31.7	52.7
+ phrase level	40.8	43.6	32.6	49.9
+ word + phrase level	41.1	44.4	33.6	48.7

Table 5: Oracle machine translation results, assuming all mouthings were recognized correctly

On the phrase level, we add binary as well as count features to the phrase table, indicating whether a gloss with a certain mouthing is translated into the corresponding spoken word (boolean feature) or counting the number of glosses in the phrase for which this is true (count feature).

Thus, the computer would e.g. learn to translate the gloss variant MOUNTAIN_alps (which consists of the manual sign for mountain, accompanied by the mouthing "Alps") into the German word for Alps. We refrained from hard-wiring these connection for two reasons. First, the viseme recognition also contains errors, which can partly be learnt by the machine translation system during training. Moreover, mouthings usually use the base form of the word without inflections, and thus the same mouthing can result in different inflections in the spoken language.

First we examine oracle translation results which assume that all mouthings have been recognized correctly. These results form an upper bound on the translation performance of the actual system and show the potential of adding the mouthing information to the system. The results can be seen in Table 5. Training a phrase-based system on the gloss-variants increases the system performance by 6 BLEU and 6.4 TER. Additional gains can be optained by increasing the probabilities of matching mouthings and translations on the word and phrase level. The best performance can be obtained by combining both of these models.

The translation result of the whole pipeline of viseme recognition and translation system is given in Table 6. Training the machine translation system on the gloss variants produced by the viseme recognizer leads to a degragation in BLEU, but TER is improved. Increasing the weight of corresponding mouthing and translation pairs either on the word or the phrase level leads to an improvement. Combining both models only slightly improves the BLEU score.

7. Conclusions / Outlook

In this paper, we propose the integration of a viseme recognizer into a sign language translation framework. Instead of using the facial features in the recognition phase, we opt for using the mouthing information as an additional knowledge source in the translation system. The system is able to out-

System	Dev		Test	
	BLEU	TER	BLEU	TER
Baseline System	35.5	58.8	23.8	66.5
Viseme + MT System	35.2	53.2	23.1	65.4
+ word level	36.1	54.3	24.1	65.5
+ phrase level	36.8	53.5	24.4	64.4
+ word + phrase level	37.5	52.6	24.8	64.4

Table 6: Machine translation results of systems including viseme recognition input

perform the baseline system which only translates the manual information of the signs. The use of mouthing information is especially useful in countries which have an oralist education tradition. In other countries, e.g. the US, fingerspelling is used more heavily.

In the future, we want to improve the quality of the viseme recognition by including a histogram of the mouth area. This can lead to improvements for visemes with distinct tongue or teeth configurations. Moreover, we want to incorporate other modalities besides the hands and the mouthing as well. One problem which we encountered during the experiments is the spreading of the mouthing, i.e. the mouthing is not synchronous to the hands but starts later. We want to address this issue using dynamic time alignment.

8. References

[1] I. Ari, A. Uyar, and L. Akarun, "Facial feature tracking and expression recognition for sign language," in *Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on*, 2008, pp. 1–6.

[2] I. Rodomagoulakis, S. Theodorakis, V. Pitsikalis, and P. Maragos, "Experiments on global and local active appearance models for analysis of sign language facial expressions," in *9th International Gesture Workshop on Gestures in Embodied Communication and Human-Computer Interaction*, 2011, pp. 96–99.

[3] J. Piater, T. Hoyoux, and W. Du, "Video analysis for continuous sign language recognition," in *Proceedings of 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 2010, pp. 22–23.

[4] U. Von Agris, J. Zieren, U. Canzler, B. Bauer, and K.-F. Kraiss, "Recent developments in visual sign language recognition," *Universal Access in the Information Society*, vol. 6, no. 4, pp. 323–362, 2008.

[5] S. Werda, W. Mahdi, and A. B. Hamadou, "Lip localization and viseme classification for visual speech recognition," *CoRR*, vol. abs/1301.4558, 2013.

[6] S. Morrissey and A. Way, "Manual labour: tackling machine translation for sign languages," *Machine Translation*, vol. 27, no. 1, pp. 25–64, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10590-012-9133-1

[7] T. Johnston, "The lexical database of auslan (australian sign language)," *Sign Language and Linguistics*, vol. 4, pp. 145–169(25), 2001.

[8] J. Forster, C. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. Piater, and H. Ney, "Rwth-phoenix-weather: A large vocabulary sign language recognition and translation corpus," in *International Conference on Language Resources and Evaluation*, Istanbul, Turkey, May 2012. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/844_Paper.pdf

[9] D. Stein, C. Schmidt, and H. Ney, "Analysis, preparation, and optimization of statistical sign language machine translation," *Machine Translation*, vol. 26, no. 4, pp. 325–357, Dec. 2012. [Online]. Available: http://dx.doi.org/10.1007/s10590-012-9125-1

[10] G. J. Edwards, C. J. Taylor, and T. F. Cootes, "Interpreting face images using active appearance models," in *Proc. International Conference on Automatic Face and Gesture Recognition*, June 1998, pp. 300–305.

[11] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[12] R. Gross, I. Matthews, and S. Baker, "Generic vs. person specific active appearance models," *Image and Vision Computing*, vol. 23, no. 12, pp. 1080–1093, 2005.

[13] J. Xiao, S. Baker, I. Matthews, and T. Kanade, "Real-time combined 2d+ 3d active appearance models," in *Proc. Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. II–535.

[14] B. Aschenberner and C. Weiss, "Phoneme-viseme mapping for german video-realistic audio-visual-speech-synthesis."

[15] D. Rybach, C. Gollan, G. Heigold, B. Hoffmeister, J. Lööf, R. Schlüter, and H. Ney, "The rwth aachen university open source speech recognition system," in *Interspeech*, Brighton, UK, Sept. 2009, pp. 2111–2114.

[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, p. 381395, 1981.

[17] D. Vilar, D. Stein, M. Huck, and H. Ney, "Jane: Open Source Hierarchical Translation, Extended with Reordering and Lexicon Models," in *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden, July 2010, pp. 262–270.

The AMARA Corpus: Building Resources for Translating the Web's Educational Content

Francisco Guzman, Hassan Sajjad, Stephan Vogel, Ahmed Abdelali

Qatar Computing Research Institute
Qatar Foundation
{fguzman,hsajjad,svogel,aabdelali}@qf.org.qa

Abstract

In this paper, we introduce a new parallel corpus of subtitles of educational videos: the AMARA corpus for online educational content. We crawl a multilingual collection community generated subtitles, and present the results of processing the Arabic–English portion of the data, which yields a parallel corpus of about 2.6M Arabic and 3.9M English words. We explore different approaches to align the segments, and extrinsically evaluate the resulting parallel corpus on the standard TED-talks tst-2010. We observe that the data can be successfully used for this task, and also observe an absolute improvement of 1.6 BLEU when it is used in combination with TED data. Finally, we analyze some of the specific challenges when translating the educational content.

1. Introduction

Lecture Translation has become an active field of research in the wider area of Speech Translation [1, 2]. This is demonstrated by large scale projects like the EU-funded translectures [3] and by evaluation campaigns like the one organized as part of the International Workshop on Spoken Language Translation (IWSLT), which introduced the challenge to translate TED talks [4] for the 2010 competition. However, the main limitation for the success of these projects continues to be the access to high quality training data.

With the emergence of Massive Online Open Courses (MOOCs), thousands of video lectures have already been generated. Sites like Khan Academy[1], Coursera[2], Udacity[3], etc., continuously increase their repertoire of lectures, which range from basic math and science topics, to more advanced topics like machine learning, also covering history, economy, psychology, medicine, and more.

Online education has bridged the geographical and financial gap, enabling students to access high quality content for free, irrespective of their location. However, the access to this content is still limited by language barriers. By far the most content available is in English. This severely limits access to this high-quality educational material for learners not being able to read and understand English. To overcome these language barriers, amazing efforts are undertaken by volunteers, to translate such lectures into many other languages. One example is the already mentioned TED Talks[4], for which so far more then $9,000$ volunteers have generated about $40,000$ translations into a total of 101 languages. While this and similar efforts at Khan Academy or MIT's Open Courseware[5] are highly commendable, the coverage is extremely skewed towards a small number of languages. It is therefore clear that manual translation trails behind, and that for many languages the small number of volunteers cannot keep up with the fast pace in which new content is appearing on these educational platforms.

Statistical machine translation (SMT) can bridge this gap by automatically translating videos for which subtitles are not available. It also can support volunteer translators, by providing an initial translation, which then can be post-edited [5]. Thus, SMT has the potential to increase the penetration of educational content, allowing it to reach a wider audience. To achieve this, an SMT system requires a large quantity of high-quality in-domain training data. Unfortunately, large data for machine translation has traditionally been constrained to domains such as legal documents, parliamentary proceedings and news. So far, the only openly accessible corpus for the lecture domain has been the TED talks [6].

In this paper, we introduce a new parallel corpus of subtitles of educational videos: the AMARA corpus for online educational content. We crawl a collection of multilingual community-generated subtitles[6]. Furthermore, we explore the steps necessary to build corpora suitable for Machine Translation by processing the Arabic-English part of the multilingual collection. This yields a parallel corpus of about 2.6M Arabic and 3.9M English words. We explore different approaches to align the subtitles, and verify the quality of the generated parallel corpus by building translation models, and extrinsically evaluating them on the standard TED-talks tst-2010 from IWSLT 2011, and on our proposed AMARA test set. We show that the AMARA corpus shares similar domain with TED-talks and leads to an increase of translation quality on the TED translation task.

[1] https://www.khanacademy.org/
[2] https://www.coursera.org/
[3] https://www.udacity.com/
[4] http://www.ted.com/
[5] http://ocw.mit.edu/index.htm
[6] Publicly available through the Amara website: http://www.amara.org

In the next section,we describe the related work and in Section 3 we present crawling, segmentation and statistics of the AMARA corpus. Section 4 shows the usability of AMARA alone and combined with IWSLT for machine translation. In Section 5, we present error analysis based on machine translation output. Section 6 presents our conclusions and future work.

2. Related Work

Several corpora have been developed to support the seminar and lecture translation efforts. One example is the corpus form Computers in the Human Interaction Loop (CHIL) [7], which consists of recordings and transcriptions of technical seminars and meetings in English. The content of the corpus includes a variety of topics: from audio and visual technologies to biology and finance. It is available through ELRA[7] to its members.

More recently, the IWSLT10 [4] evaluation campaign has turned its attention to the lecture and seminar domain by focusing on TED talks. To support this task, a collection of lecture translations has been automatically crawled from the TED website in a variety of languages and made publicly available through the WIT[3] project [8]. In this paper, we used such data as a point of comparison. We crawl parallel subtitles of educational videos and use several measures to show the quality of the crawled corpus in comparison with the closely related IWSLT data set.

In the past, multilingual corpora creation from user-contributed movie subtitles has been addressed by [9]. Recently, a large collection of parallel movie subtitles from the Opensrt[8] community along with tools for alignment of these has been made available through the Opus project [10].

Combination of corpora to improve the translation model has been explored with relative success in the past. For the NewsCommentary and OpenSrt corpora, [11] explore different ways to mix the phrase-table to adapt the Europarl corpus. For the Arabic-English IWSLT data, [12] achieve a relative improvement of 0.7 BLEU by mixing phrases from UN and IWSLT data using instance weighting with weights coming from the language model perplexity.

In this paper, we present the experimental results from data gathered from publicly available crowd-generated data, that has proved to be useful for the lecture domain, but that poses specific challenges, as it has a special focus on online education.

3. The AMARA Corpus

Amara is a web-based platform for editing and managing subtitles of online videos. It provides an easy-to-use interface, which allows users to collaboratively subtitle and translate those videos. The site uses a community-refereed approach to ensure the quality of the transcriptions and translations in the spirit of Wikipedia.

[7] www.elra.org

[8] www.opensrt.org

Amara works in collaboration with online educational organizations like KhanAcademy, TED, and Udacity. As a result, a large body of translations of educational content is available in multiple languages. For example, for Udacity, more than 25K subtitles for over 10K videos have been created by a team of 917 volunteers, since December 2012. These translations are publicly accessible through the Amara website in the form of downloadable video subtitles.

3.1. Languages

On the Amara website, the number of different languages into which a video has been subtitled varies from video to video. In Figure 1 we observe the overall distribution of the number of available languages per video by the total number of videos on the Amara website having translations available in that many languages. A few videos have subtitle translations in as many as 109 different languages. Furthermore, at least 1000 videos have translations available in 25 different languages, and 3000 have translations available in at least 6 different languages. However, the distribution quickly tails off, as many videos have been translated into only a few languages.

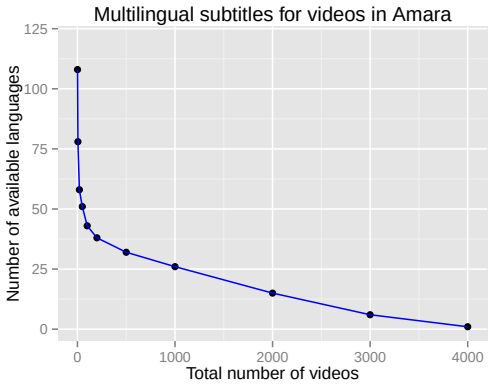


Figure 1: Distribution of the number of available languages per video by the total number of videos in the Amara website.

The most represented languages in the subtitles of this repository are: English with 90K subtitles, French with 20K subtitles, Spanish with 20K subtitles, Italian with 8.8K subtitles and Arabic with 5.9K subtitles. On the other hand, the original language of the videos is highly dominated by English with 135K videos, followed by Spanish with 8.7K videos, French with 6.1K videos, German with 5.0K videos and Russian with 4.3K videos.

In Table 1 we present the distribution of videos from different languages that have been translated into English and Arabic. We observe that English is by far the most subtitled language, which should not be a surprise given the large number of available videos in the platform. Still, only 39% of all the English videos are subtitled into English. However, Arabic videos have an unusually high number of translations into English. In fact, for Arabic videos, there are more English subtitles than Arabic subtitles, which means that many

Videos		Subtitled into	
Language	Total	Arabic	English
English	135K	4463	54023
Arabic	3.8K	494	1286
Spanish	8.7K	33	1167
French	6.1K	38	1160
German	5.0K	11	1006

Table 1: Distribution of the number of translations into Arabic and English from the most popular video languages in the Amara platform. As of December 1st, 2013

videos are translated directly into English, without taking the route through generating Arabic subtitles first. At this point, about 33% of all Arabic videos are subtitled into English, which is a larger proportion when compared to Spanish (13%), French (19%) and German (20%). Note that this data could possibly mislabeled and contain wrong language information. Noisy data often results in poor word alignments and weak translation models.

To shed light on how valuable this data can be for machine translation, we examine the impact of the Arabic-English collection of subtitles, that we codename the AMARA Corpus, in a machine translation environment. These represent only a small fraction of the data available on the Amara website. In future, we plan to extend our work to other language pairs.

3.2. Crawling

The Amara site provides a list of videos and the number of languages the media has been subtitled into. Additionally, it allows filtering by languages. This resulted in 4338 videos that have subtitles in both English and Arabic[9]. In most cases, the original language of these videos is English. Using a non-intrusive in-house crawler, and in cooperation with amara.org, we collected the subtitle files for both Arabic and English. In the current version of the data, we did not perform any additional validation to verify that the documents are in the language they claim to be. Instead, we perform an indirect measurement of the quality by using the parallel data for a standard Machine Translation task.

The subtitle files are in Sub-Rip Text file format (.srt). It consists of segments that are formed by three components:

Segment ID: A number, in sequence, identifying the segment.
Time interval: The start and end times of the subtitle, which represent the timeframe the particular subtitle appears on the screen.
Content: The text for the subtitle segment, with one or more lines.

[9]This quantity includes videos originated in any language pair, not only Arabic and English. The date of collection was July 1st, 2013.

3.3. Data Filtering

From the crawled data for the Arabic-English language pair, we obtained subtitles for a total of 4338 videos, which originated from different organizations. These subtitle files also included transcriptions for the TED talks. To assess the usefulness of this data for translating a standard set for lecture translation such as the IWSLT-11 dataset, we decided to exclude all possible overlap with the IWSLT talk data to avoid contamination and thereby overly optimistic results. Unfortunately, the AMARA data does not have extensive meta-data that can be used for document-level filtering. Furthermore, the difference in sentence alignments, tokenization between our data and the IWSLT-talk data also posed a challenge.

To handle tokenization differences, we detokenized AMARA documents and re-tokenized them using the identical scheme as used for IWSLT. Furthermore, we calculated the percentage of overlap between each of the AMARA documents, and the IWSLT data (train, tune and test); and filtered out the documents ones that presented an overlap of more than a certain threshold (in this case 1% of the sentences in the document). However, due to the conversational nature of the data, frequent phrases such as "applause", "thank you", etc., match almost every document. As a consequence, the relative overlap of smaller documents was artificially inflated and they were filtered out. We fixed this by applying a strong constraint that prevented duplicated counts. Therefore, once a sentence from a specific document was matched with the IWSLT data, it could not be matched to any other document. Our assumption here is that there are no redundant documents in the pool of AMARA documents, so removing previously matched sentences would not cause any trouble. We tested filtering both with and without deduplication. In practice, there were not major differences between the two generated corpora. Thus, we kept the one with the strong constraint, which generated 2400 bilingual documents.

3.4. Segment Alignment

The collected subtitles are for the most part, parallel at the segment level. About 75% percent of all collected segments have identical time stamps on both sides. However, there are two cases, which lead to non-parallel segments:

Incomplete data: When the data in one language (mostly Arabic) is not complete. This could be the case when the translation is still in progress.
Different timestamps: When the text of source and target segment correspond to each other, but the timestamps are not synchronized across languages. This happens when the subtitles in the second language are not generated by translating the subtitles in the original language, but done directly by listening to the original sound track, and translating on the fly.

In order to deal with these issues, we used several algorithms to align the subtitle files. Below, we briefly summarize them:

Strict synchronization constraint (Baseline)
We only extracted the segments from the parallel files if they have identical segment IDs and timestamps. This is a strong constraint, yet gives a good notion of how much data is truly parallel at the segment level.

Automatic sentence alignment
This approach extends the assumption that translations tend to be similar in length [13] by using information from a bilingual dictionary to improve the alignment between parallel files. We used the implementation provided by Hunalign [14]. It aligns the parallel text in two passes.

First, sentence length and lexicon (if provided) information is combined to perform an initial alignment. A new, corpus specific lexicon is then generated from the resulting word alignment. A second pass is performed to align the text with the newly generated dictionary. Note that this approach allows merging of multiple consecutive segments into one longer segment.

Subtitle synchronization
This approach, as implemented in the Uplug subtitle alignment tool [10], exploits the timing information available in the subtitles to perform the alignment. It assumes that sentences that appear in close time-frames should be closer to each other. It can be enhanced by providing anchor-points from which timing offsets and speed ratios can be resolved [9].

The alignment can be enhanced by a bilingual dictionary or by exploiting cognates (LCSR) to establish better anchor points. To synchronize segments across different time-frames, this approach can merge several input segments into one output sentence.

Cascaded synchronization
This approach is a combination of the first two approaches. We started by enforcing a strict synchronization constraint on different subtitles. Then we performed word alignment on the concatenation of all of the strictly aligned data, and extracted a lexicon from the resulting alignment. This lexicon was then used to run the automatic sentence aligner on the unsynchronized portions of the subtitles. Finally, we concatenated both the strictly synchronized with the automatically aligned portions of the subtitles.

3.5. Synchronization Results

Table 2 presents the corpus statistics for the different parallel corpora resulting from the different alignment approaches. The strict synchronization loses a significant portion of the overall data, as shown by the lower total number of words. The segments are short, with only 9.4 words per segment.

	Corpus Statistics		
Algorithm	pairs	tokens	types
Strict Sync	306K	2.9M	55.2K
Hunalign	223K	3.9M	58.2K
Uplug+Cog	221K	3.9M	58.2K
Uplug+Dict	221K	3.9M	58.2K
Uplug+Cog+Dict	221K	3.9M	58.2K
Cascaded	382K	3.6M	58.2K
IWSLT11	93K	1.8M	43.1K

Table 2: Corpus statistics and translation results for different sentence alignment algorithms: strict synchronization (Strict Sync), automatic sentence alignment (Hunalign), subtitle synchronization (Uplug), and cascaded sentence alignment. IWSLT11 shows the statistics of the IWSLT 2011 data.

The sentence aligner (Hunalign) and all the variants of synchronization algorithm (Uplug) yield very similar results in terms of number of words and vocabulary size. However, the segments are now much longer, about 17 words per segment, showing that indeed, Uplug and Hunalign collapse different segments into one sentence pair.

The cascaded alignment preserves the original segment length (9.4 words), while diminishing the loss of tokens. Shorter sentence pairs typically yield better word alignment, which should help to improve the translation quality. On the other side, segmenting sentences into shorter segments means that longer phrases cannot be extracted, which would be extracted from concatenated segments. Segmentation for speech translation has been studied in the past, with somewhat conflicting results [15, 16] and needs to be revisited.

Despite observing a similar performance between all the synchronization variants, for the remainder of this paper we will use the corpus resulting from the cascaded synchronization alignment.

4. Experimental Results

In this section, we extrinsically evaluate the usefulness of the AMARA corpus by training models the data, and observing its performance on a IWSLT lecture translation task (2011). We explore different adaptation methods to better utilize the AMARA data for the IWSLT talk translation task.

4.1. Datasets

To evaluate the usefulness of the crawled data, we experimented with the Arabic-English datasets from the IWSLT 2011 Evaluation Campaign[6]. The IWSLT dataset contained train, dev-2010 and tst2010 sets which consist of 90.5K , 934, 1.6K parallel sentences respectively. In these experiments, we did not make use of the additional IWSLT monolingual data, i.e. the language models in most experiments use only the English side of the parallel corpora, but we also report results using a GigaWord LM.

We used the AMARA corpus resulting from the cascaded synchronization. We divided this corpus into several datasets by randomly sampling the available subtitles. This generated 370K, 5K, 3.6K and 4.4K sentences to be used for train, tune, test and a second test set[10], respectively.

We used IWSLT dev-2010 set for tuning and then tested on two datasets: the IWSLT tst-2010 and AMARA tst-2013, each with a single reference translation. This allowed us to benchmark the improvements obtained by using the AMARA corpus with a standard test set (the former), and to gain insights about translating online educational data (the latter).

In Table 3 we present the 5-gram, Kneser-Ney smoothed, open-vocabulary language-model perplexity for the target side of the test sets given the training corpora. Observe that while the IWSLT10 has similar perplexity w.r.t. the AMARA and IWSLT language models, the reverse relationship does not hold. The AMARA test data has a broader domain, which is not fully captured by the IWSLT language model, which is limited to TED lectures.

training LM	**testset**			
	AMARA13		IWSLT10	
	PPL	OOV	PPL	OOV
AMARA	107.5	1.3	116.7	1.6
IWSLT	204.5	2.6	107.7	1.5

Table 3: Target side per word perplexity (PPL) and out-of-vocabulary rate (OOV %) of the test sets with respect to the language model built on the training data

4.2. Experimental Setup

Preprocessing: We tokenized the English side of all bi-texts as well as the monolingual data (GigaWord) for language modeling using the standard tokenizer of the Moses toolkit [17]. We further truecased this data by changing the casing of each sentence-initial word to its most frequent casing in the training corpus. For the Arabic side, we segmented the corpus following the ATB segmentation scheme with the Stanford word segmenter [18].

Training: We built separate directed word alignments for English→Arabic and for Arabic→English using IBM model 4 [19], and symmetrized them using *grow-diag-final-and* heuristic [20]. We extracted phrase pairs of maximum length seven. We scored these phrase pairs using maximum likelihood with Kneser-Ney smoothing,as implemented in the moses toolkit, thus obtaining a phrase table where each phrase-pair has the standard five translation model features. We also built a lexicalized reordering model : *msd-bidirectional-fe*. For language modeling, we trained a separate 5-gram Kneser-Ney smoothed LM model on each available corpus (target side of a training bi-text or monolingual dataset) using KenLM [21]; we then interpolated these models minimizing the perplexity on the target side of the tuning dataset (IWSLT dev-2010). Finally, we built a large joint log-linear model, which used standard SMT feature functions: language model probability, word penalty, the parameters from the phrase table, and those from the reordering model.

We used the phrase-based SMT model as implemented in the Moses toolkit [17] for translation, and reported evaluation results over two datasets. We reported BLEU calculated with respect of the original reference using NIST v13a, after detokenization and recasing of the system's output.

Tuning: We tuned the weights in the log-linear model by optimizing BLEU [22] on the tuning dataset, using PRO [23] with the fixed BLEU prosposed by [24]. We allowed the optimizer to run for up to 10 iterations, and to extract 1000-best lists for each iteration.

Decoding: On tuning and testing, we used monotone-at-punctuation decoding (this had no impact on the translation length). On testing, we further used cube pruning.

4.3. Baseline B_1

For the baseline system, we trained the phrase and the reordering models on the IWSLT training dataset. The language model was trained on the English side of the IWSLT training data. We tuned the weights on IWSLT-dev2010. Below, we present the experimental results when using the AMARA data for the translation model, the language model and both.

4.4. AMARA Data and the Translation Model

We investigated several ways to maximize the impact of the AMARA corpus for translation by building variations of the translation and reordering models. The systems presented in this section used the same language model built on the English side of the IWSLT training data. As for the baseline, the weights are tuned on the IWSLT-dev2010. Following are different translation settings that we experimented with.

AMARA only (TM_1): Instead of using the IWSLT training data, we built the translation and reordering models using only the AMARA corpus.

Concatenation (TM_2): In this setting, we concatenated AMARA with IWSLT for training of the translation and reordering models. This generally improves word alignment, reduces OOV rate and improves translation quality if two corpora are from similar domain. However, if the added corpus is noisy or of out-of-domain, (e.g. UN data), we can observe a degradation in performance.

Phrase table combination (TM_3): We applied phrase table combination as described in [25]. We built two phrase tables and reordering models separately on the IWSLT and AMARA data. Then, we merged them by adding three additional indicator features to each entry to inform the decoder if the phrase was found in the first, second or both tables. This can be seen as a form of log-linear interpolation.

[10]We did not use the second test set for the experiments in this paper.

SYS	TM	IW10	OOV	AM13	OOV
B_1	IWSLT	22.97	1.9	23.26	3.9
TM_1	AMARA	22.40	2.4	23.66	1.7
TM_2	IW+AM	23.41	1.2	27.63	1.8
TM_3	PT(IW,AM)	**23.57**	1.2	**27.65**	1.8

Table 4: Results of the translation system tested on IWSLT-tst2010 and AMARA-tst2013. All systems use identical language model built on the IWSLT training data and use IWSLT-dev2010 for tuning.

4.4.1. Results

Table 4 shows the results of using the different translation models. Using only AMARA for translation model (TM_1) showed competitive results with our baseline B_1 that is built on IWSLT data. The comparable BLEU score on IWSLT10 shows the value of the AMARA corpus as a parallel corpus in the IWSLT10 translation task. Furthermore, the concatenation and merging of AMARA and IWSLT are able to further reduce the OOV rate. From these combinations, we observe a BLEU improvement up to 0.6 for IWSLT10 and 4.4 for AMARA[11].

4.5. AMARA Data and the Language Model

In this section, we explore the usability of the AMARA data for language modeling. For every system, the translation and reordering models were trained on the IWSLT data and tuned on IWSLT-dev2010. We experimented with different approaches to build the language models:

AMARA only (LM_1): used a LM trained exclusively on the target side of the AMARA corpus.

Concatenation (LM_2): used a concatenation of the English side of both the IWSLT and AMARA corpora.

Interpolation (LM_3): used an interpolated from B_1 and LM_1. The interpolation weights were set to minimize perplexity on the target side of IWSLT-dev2010.

Gigaword (LM_4): uses LM built on the English GigaWord (v5) corpus. This was only included as a reference.

4.5.1. Results

Table 5 summarizes the results of our experiments. Using only AMARA for language model slightly hurts the performance on IWSLT10 by 0.14 BLEU points. However, it has better results when tested on AMARA13. Both the concatenated and interpolated language models show improvements in the translation quality of both sets.

4.6. Best Combination

We combined the best translation model and language model settings from Table 4 and Table 5 respectively and summarize the results in Table 6. From these results we can observe that using AMARA data with IWSLT gives up to a 1.69 improvement in BLEU for the IWSLT-tst2010 and 8.84 BLEU for the AMARA-tst2013. While the results on the AMARA set might seem unrealistically high, we need to remember that the IWSLT baseline is out-of-domain for the AMARA test set, as explained by the high perplexity in table 3. Improving an out-of-domain baseline with in-domain data with translation model adaptation has been observed to give such high jumps in performance [11].

[11]The higher gain in BLEU for AMARA13 might be an artifact of using IWSLT target side for LM and IWSLT-dev for tuning.

SYS	LM	IW10	AM13
B_1	IWSLT	22.97	23.26
LM_1	AMARA	22.83	24.05
LM_2	IWSLT+AMARA	**23.69**	**25.90**
LM_3	INTERPOL	23.59	25.62
LM_4	GW	24.24	24.79

Table 5: Results of the translation system tested on IWSLT-tst2010 and AMARA-tst2013. All systems use identical translation model built on the IWSLT training data and use IWSLT-dev2010 for tuning.

SYS	TM	LM	IW10	AM13
B_1	IWSLT	IWSLT	22.97	23.26
S_1	TM_3	LM_3	**24.66**	31.62
S_2	TM_2	LM_2	24.33	**32.10**

Table 6: Results of the translation system tested on IWSLT-tst2010 and AMARA-tst2013. S_1 uses interpolated language model and merged phrase table to build translation model. S_2 uses concatenated training data for both translation model and language model.

In summary, we observed that both in isolation and in combination, the parallel and monolingual data from the volunteer-fueled AMARA corpus, is of sufficient quality to be used for a lecture translation task.

5. Error Analysis

For this section, we analyzed the errors performed during the translation of the AMARA13 testset. This was done to determine what are the specific challenges found when translating this set. We further provide a brief discussion of ways in which these problems can be fixed in the future. To do so, we classify the most important errors in two categories:

5.1. Mathematical quantifiers and numbers

One specific case of problem where recall is particularly low, refers to the translation of certain mathematical forms and numbers. This phenomenon is observed in instances where the numbers and operations were spelled out in the English side while in Arabic they are provided in their mathematical notation. For instance, the expression “is equal to” had a recall of 0 out of 41 times. The “the derivative of” was correctly translated only 6 out of 23 times. These problems

arise from the non-homogeneity with which mathematical texts are translated. For example:

Ar: ومرة اخرى هذا يساوي $+3+2$ ويساوي 5
En: Once again that's two plus plus three, so that equals five.

Ar: نحن بحاجة لتقييم نهاية اقتراب x من ما لا نهاية لـ $4x \hat{} 2 - 5x$ ، وكل ذلك مقسوم على $1 - 3x \hat{} 2$
En: We need to evaluate the limit, as x approaches infinity, of 4x squared minus $5x$, all of that over 1 minus $3x$ squared .

We observe that on the Arabic side, the mathematical symbols and digits are preferred, while in English, these are spelled out. A similar problem is the text-to-number conversion, which has been previously solved using rule-based approaches. In this case, a more refined set of rules can be devised to homogenize mathematical notation on both the source and target side of the corpus.

5.2. OOVs and transliteration

OOVs from languages with different scripts pose a challenge for readability. In an educational context, these need to be minimized and dealt correctly.

In the AMARA set, we observed that English terms are sometimes used in Arabic to denote English named entities. Examples of such cases are: Nevis, Yukon, Blanc, which are names of mountains used for math problems. These words can be left "untranslated" and the issue will be resolved.

A different problem, specific to Arabic-to-English translation, particularly for the technical domain, is the occurrence of OOVs related to neologisms. Fortunately many of these can be tackled by simple transliteration. For instance: جافاسكربت (javascript), ميدياغوبلن (media goblin), وركرافت (Warcraft), etc.

Together, these two problems account for 8 of the top 10 most frequent OOVs, this represents at least 12% of all the OOV words found in the testset.

6. Conclusion and Future Work

In this paper, we used data generated by a community of volunteers to advance the state-of-art of machine translation for educational content. This data, available through the AMARA platform, provides an opportunity to build a large, multilingual corpus, which can help to provide automatic translations in cases where no manual translation is available.

At this time, we explored the Arabic-English parallel portion of the data, and we evaluated its usefulness by translating the TED task of the IWSLT data. We presented different ways to process the data, especially to deal with problems in the original segment alignment. We showed that this data can be successfully used to translate lectures.

In addition, we used a new test set with AMARA specific data, geared towards educational translation. We observed that this data covers a broader domain than the IWSLT, and has specific challenges, some of which we analyzed. For instance, stylistic preferences when translating mathematical expressions, are prevalent and crucial for the content to be translated correctly.

In the future, we plan to extend the processing of the AMARA corpus to include at least 25 languages. Adding meta-data, like domain and topic, speaker, transcriber, and translator IDs, will allow using this corpus for speech translation research. For example, studying model adaptation or developing translation strategies to deal with the specific language and notation used in mathematics, biology, chemistry, etc. Finally, we plan to leverage the social graph of volunteers to be able to assign confidence to their translations depending on their characteristics (e.g. number of translations completed, domain of expertise, etc.). In summary, this data presents many possible lines of research. We are currently evaluating the different alternatives to make this corpus publicly available, while respecting copyright.

7. Acknowledgments

We would like to thank Nicholas Reville and the Amara staff for their support.

8. References

[1] C. Fügen, M. Kolss, D. Bernreuther, M. Paulik, S. Stücker, S. Vogel, and A. Waibel, "Open domain speech recognition & translation: Lectures and speeches," in *Acoustics, Speech and Signal Processing*, ser. ICASSP '06, 2006.

[2] C. Fügen, A. Waibel, and M. Kolss, "Simultaneous translation of lectures and speeches," *Machine Translation*, vol. 21, no. 4, pp. 209–252, 2007.

[3] J. A. Silvestre-Cerdà, M. A. del Agua, G. Garcés, G. Gascó, A. Giménez, A. Martínez, A. Pérez, I. Sánchez, N. Serrano, R. Spencer, J. D. Valor, J. Andrés-Ferrer, J. Civera, A. Sanchis, and A. Juan, "TransLectures," in *Online Proceedings of Advances in Speech and Language Technologies for Iberian Languages*, ser. IBERSPEECH '12, Madrid, Spain, 2012.

[4] M. Paul, M. Federico, and S. Stüker, "Overview of the IWSLT 2010 evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, ser. IWSLT '10, 2010.

[5] S. Green, J. Heer, and C. D. Manning, "The efficacy of human post-editing for language translation," in *ACM Human Factors in Computing Systems*, ser. CHI '13, 2013.

[6] M. Federico, S. Stüker, L. Bentivogli, M. Paul, M. Cettolo, T. Herrmann, J. Niehues, and G. Moretti, "The IWSLT 2011 evaluation campaign on automatic talk

translation," in *Proceedings of the Eight International Conference on Language Resources and Evaluation*, ser. LREC '12, Istanbul, Turkey, 2012.

[7] D. Mostefa, N. Moreau, K. Choukri, G. Potamianos, S. M. Chu, A. Tyagi, J. R. Casas, J. Turmo, L. Cristoforetti, F. Tobia, *et al.*, "The CHIL audiovisual corpus for lecture and meeting analysis inside smart rooms," *Language Resources and Evaluation*, vol. 41, no. 3-4, pp. 389–407, 2007.

[8] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation*, ser. EAMT '12, Trento, Italy, 2012.

[9] J. Tiedemann, "Synchronizing translated movie subtitles," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, ser. LREC '08, 2008.

[10] J. Tiedemann, "Parallel data, tools and interfaces in OPUS," in *Proceedings of the Eight International Conference on Language Resources and Evaluation*, ser. LREC '12, 2012.

[11] B. Haddow and P. Koehn, "Analysing the effect of out-of-domain data on SMT systems," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*, ser. WMT '12, Montreal, Canada, June 2012.

[12] S. Mansour and H. Ney, "A simple and effective weighted phrase extraction for machine translation adaptation," in *Proceedings of the International Workshop on Spoken Language Translation*, ser. IWSLT '12, 2012.

[13] W. A. Gale and K. W. Church, "A program for aligning sentences in bilingual corpora," *Computational linguistics*, vol. 19, no. 1, pp. 75–102, 1993.

[14] D. Varga, L. Németh, P. Halácsy, A. Kornai, V. Trón, and V. Nagy, "Parallel corpora for medium density languages," in *Proceedings of the Recent Advances in Natural Language Processing*, ser. RANLP '05, 2005.

[15] S. Rao, I. Lane, and T. Schultz, "Optimizing sentence segmentation for spoken language translation," in *Proceedings of International Speech Communication Association*, ser. INTERSPEECH '07, Antwerp, Belgium, 2007.

[16] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence segmentation and punctuation recovery for spoken language translation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ser. ICASSP '08, Las Vegas, Nevada, USA, 2008.

[17] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (Demonstration session).*, ser. ACL '07, Prague, Czech Republic, 2007.

[18] S. Green and J. DeNero, "A class-based agreement model for generating accurately inflected translations," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, ser. ACL '12, Jeju Island, Korea, 2012.

[19] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[20] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, ser. HLT-NAACL '03, Edmonton, Canada, 2003.

[21] K. Heafield, "KenLM: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, ser. WMT '11, Edinburgh, UK, 2011.

[22] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ser. ACL '02, Philadelphia, PA, USA, 2002.

[23] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11, Edinburgh, Scotland, United Kingdom, 2011.

[24] P. Nakov, F. Guzmán, and S. Vogel, "Optimizing for sentence-level BLEU+1 yields short translations," in *Proceedings of the 24th International Conference on Computational Linguistics*, ser. COLING '12, Mumbai, India, 2012.

[25] P. Nakov and H. T. Ng, "Improved statistical machine translation for resource-poor languages using related resource-rich languages," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '09, Singapore, 2009.

Constructing a Speech Translation System using Simultaneous Interpretation Data

Hiroaki Shimizu, Graham Neubig, Sakriani Sakti,
Tomoki Toda, Satoshi Nakamura

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma-shi, Nara, Japan
{hiroaki-sh,neubig,ssakti,tomoki,s-nakamura}@is.naist.jp

Abstract

There has been a fair amount of work on automatic speech translation systems that translate in real-time, serving as a computerized version of a simultaneous interpreter. It has been noticed in the field of translation studies that simultaneous interpreters perform a number of tricks to make the content easier to understand in real-time, including dividing their translations into small chunks, or summarizing less important content. However, the majority of previous work has not specifically considered this fact, simply using translation data (made by translators) for learning of the machine translation system. In this paper, we examine the possibilities of additionally incorporating simultaneous interpretation data (made by simultaneous interpreters) in the learning process. First we collect simultaneous interpretation data from professional simultaneous interpreters of three levels, and perform an analysis of the data. Next, we incorporate the simultaneous interpretation data in the learning of the machine translation system. As a result, the translation style of the system becomes more similar to that of a highly experienced simultaneous interpreter. We also find that according to automatic evaluation metrics, our system achieves performance similar to that of a simultaneous interpreter that has 1 year of experience.

1. Introduction

While the translation performance of automatic speech translation (ST) has been improving, there are still a number of areas where ST systems lag behind human interpreters. One is accuracy of course, but another is with regards to the speed of translation. When simultaneous interpreters interpret lectures in real time, they perform a variety of tricks to shorten the delay until starting the interpretation. There are two main techniques. The first technique, also called the *salami technique*, is to divide longer sentences up into a number of shorter ones, resulting in a lower delay [1]. The second technique is to adjust the word order of the target language sentence to more closely match the source language, especially for language pairs that have very different grammatical structure. An example of this that we observed in our data of English-Japanese translation and simultaneous interpretation is shown in Figure 1. When looking at the source and the translation, the word order is quite different, reversing two long clauses: A and B. In contrast, when looking at the source and the simultaneous interpretation, the word order is similar. If a simultaneous ST system attempts to reproduce the first word order, it will only be able to start translation after it has received the full "A because B." On the other hand, if the system is able to choose the word order closer to human interpreters, it can begin translation after "A," resulting in a lower delay.

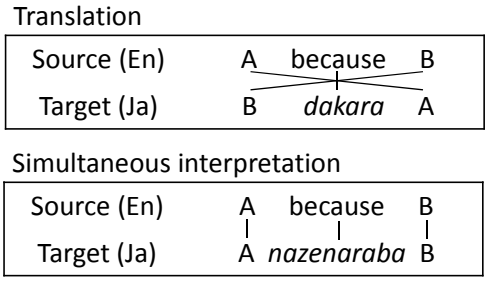


Figure 1: Difference between translation and simultaneous interpretation word order

There are several related works about simultaneous ST [2][3][4] that automatically divide longer sentences up into a number of shorter ones similarly to the salami technique employed by simultaneous interpreters. While these related works aim to segment sentences in a similar fashion to simultaneous interpreters, all previous works concerned with sentence segmentation have used translation data (made by translators) for learning of the machine translation system. In addition, while there are other related works about collecting simultaneous interpretation data [5][6][7], all previous works did not compare simultaneous interpreters of multiple experience levels and did investigate whether this data can be used to improve the simultaneity of actual MT systems.

In this work, we examine the potential of simultaneous interpretation data (made by simultaneous interpreters) to

Table 1: Profile of simultaneous interpreters

Experience	Rank	Lectures	Minutes
15 years	S rank	46	558
4 years	A rank	34	415
1 year	B rank	34	415

learn a simultaneous ST system. This has the potential to allow our system to learn not only segmentation, but also rewordings such as those shown in Figure 1, or other tricks interpreters use to translate more efficiently.

In this work, we first collect simultaneous interpretation data from professional simultaneous interpreters of three levels of experience. Next, we use the simultaneous interpretation data for constructing a simultaneous ST system, examining the effects of using data from interpreters on the language model, translation model, and tuning. As a result, the constructed system has lower delay, and achieves translation results closer to a highly experienced simultaneous interpreter than when translation data alone is used in training. We also find that according to automatic evaluation metrics, our system achieves performance similar to that of a simultaneous interpreter that has 1 year of experience.

2. Simultaneous interpretation data

As the first step to performing our research, we first must collect simultaneous interpretation data. In this section, we describe how we did so with the cooperation of professional simultaneous interpreters. A fuller description of the corpus will be published in [8].

2.1. Materials

As materials for the simultaneous interpreters to translate, we used TED[1] talks, and had the interpreters translate in real time from English to Japanese while watching and listening to the TED videos. We have several reasons for using TED talks. The first is that for many of the TED talks there are already Japanese subtitles available. This makes it possible to compare data created by translators (i.e. the subtitles) with simultaneous interpretation data. TED is also an attractive testbed for machine translation systems, as it covers a wide variety of topics of interest to a wide variety of listeners. On the other hand, in discussions with the simultaneous interpreters, they also pointed out that the wide variety of topics and highly prepared and fluid speaking style makes it a particularly difficult target for simultaneous interpretation.

2.2. Interpreters

Three simultaneous interpreters cooperated with the recording. The profile of interpreters is shown in Table 1. The most important element of the interpreter's profile is the length of their experience as a professional simultaneous interpreter. Each rank is decided by the years of experience. By comparing data from simultaneous interpretation of each rank, it is likely that we will be able to collect a variety of data based on rank, particularly allowing us to compare better translation to those that are not as good. Note that all of the interpreters work as professionals and have a mother tongue of Japanese. The number of lectures interpreted is 34 lectures for the A and B ranked interpreters, and 46 lectures for the S rank interpreter.

[1] http://www.ted.com

```
0001 - 00:44:107 - 00:45:043
本日は<H>
0002 - 00:45:552 - 00:49:206
みなさまに(F え)難しい話題についてお話したいと思います。
0003 - 00:49:995 - 00:52:792
(F え)みなさんにとっても意外と身近な話題です。
```

Figure 2: Example of a transcript in Japanese with annotation for time, as well as tags for fillers (F) and disfluencies (H)

Table 2: Translation and simultaneous interpretation data

Data		Lines	Words(EN)	Words(JA)
Translation	T1	167	3.11k	4.58k
	T2			4.64k
Simultaneous interpretation	I1			4.44k
	I2			3.67k

2.3. Transcript

After recording the simultaneous interpretation, a transcript is made from the recorded data. An example of the transcript is shown in Figure 2. The utterance is divided into utterances using pauses of 0.5 seconds or more. The time information (e.g., start and end time of each utterance) and the linguistic information (e.g., fillers and disfluencies) are tagged.

3. Difference between translation data and simultaneous interpretation data

In this section, in order to examine the differences between data created using simultaneous interpretation and time-unconstrained translation, we compare the translation data with the simultaneous interpretation data.

3.1. Setup

To perform the comparison, we prepare two varieties of translation data, and two varieties of simultaneous interpretation data. The detail about the corpus is shown in Table 2. For the first variety of translation data (T1), we had an experienced translator translate the TED data from English to Japanese without time constraints. For the second variety of translation data (T2), we used the official TED subtitles,

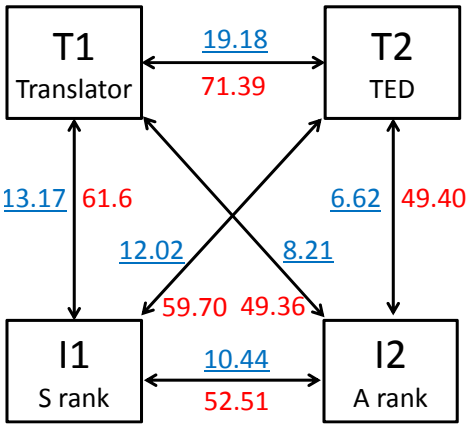


Figure 3: Results of similarity measurements between interpreters and translators. The underlined score is BLEU and the plain score is RIBES

generated and checked by voluntary translators. For the two varieties of interpretation data, I1 and I2, we used the transcriptions of the interpretations performed by the S rank and A rank interpreter respectively.

The first motivation for collecting this data is that it may allow us to quantitatively measure the similarity or difference between interpretations and translations automatically. In order to calculate the similarity between each of these pieces of data, we use the automatic similarity measures BLEU [9] and RIBES [10]. As BLEU and RIBES are not symmetric, we average BLEU or RIBES in both directions. For example, we calculate for BLEU using

$$\frac{1}{2}\{\mathrm{BLEU}(R,H)+\mathrm{BLEU}(H,R)\} \quad (1)$$

where R and H are the reference and the hypothesis. Based on this data, if the similarities of T1-T2 and I1-I2 are higher than T1-I1, T2-I1, T1-I2 and T2-I2, we can find that there are real differences between the output produced by translators and interpreters, more so than the superficial differences produced by varying expressions.

3.2. Result

The result of the similarity is shown in Figure 3. First, we focus on the relationship between the two varieties of translation data.

For T1-T2, BLEU is 19.18 and RIBES is 71.39, the highest of all in all combinations. Thus, we can say that the two translators are generating the most similar output. Next, we focus on the relationship between the translation and the simultaneous interpretation data. The similarity of T1-I1, T2-I1, T1-I2 and T2-I2 are all lower than T1-T2. In other words, interpreters are generating output that is significantly different from the translators, much more so than is explained by the variation between the translators themselves.

However, we see somewhat unexpected results when examining the relationship between the data from the two simultaneous interpreters. For I1-I2, BLEU is 10.44 and RIBES is 52.51, much lower than that of T1-T2. One of the reasons for this is the level of experience. From Table 2, we can see that the number of words translated by the A rank interpreter in I2 is almost 20 % less than that of the number of words translated by the S rank interpreter in I1. This is due to cases where the S rank interpreter can successfully interpret the content, but the A rank interpreter cannot. It is also notable that the S rank interpreter is translating almost as many words as the translation data, indicating that there is very little loss of content in the S rank interpreter's output.

However, it should be noted that I2 is more similar to I1 than either of the translators. Thus, from the view of the similarity measures used for automatic evaluation of translation, translation and simultaneous interpretation are different. Thus, in the following sections where we attempt to build a machine translation system that can generate output in a similar style to a simultaneous interpreter, we decide to evaluate our system against not the translation data, but the interpretation data of S1, which both manages to maintain the majority of the content, and is translating in the style of simultaneous interpreters.

4. Using simultaneous interpretation data

We investigate several ways of incorporating the data described in Section 2 into the MT training process.

4.1. Learning of the machine translation system

To attempt to learn a system that can generate translations similar to those of a simultaneous interpreter, we introduced simultaneous interpretation data into three steps of learning the MT system.

Tuning (Tu) : Tuning optimizes the parameters of models in statistical machine translation. The effect we hope to obtain by tuning towards simultaneous interpretation data is the learning of parameters that more closely match the translation style of simultaneous interpreters. For example, we could expect the translation system to learn to generate shorter, more concise translations, or favor translations with less reordering. In order to do so, we simply use simultaneous interpretation data instead of translation data for the development set used in tuning.

Language model (LM) : The LM has a large effect on word order and lexical choice of the translation result. We can thus assume that incorporating simultaneous interpretation data in the training of the LM will be effective to make translation results more similar to simultaneous interpretation. We create the LM using translation and interpretation data by making use of linear interpolation, with the interpolation coefficients

tuned on a development set of simultaneous interpretation data. This helps relieve problems of data sparsity that would occur if we only used simultaneous interpretation data in LM training.

Translation model (TM) : The TM, like the LM, also has a large effect on lexical choice, and thus we attempt to adapt it to simultaneous translation data as well. We adopt the phrase table by using the fill-up [11] method, which preserves all the entries and scores coming from the simultaneous interpretation phrase table, and adds entries and scores from the phrase table trained with translation data only if new.

4.2. Learning of translation timing

While in the previous section we proposed methods to mimic the word ordering of a simultaneous interpreter, our interpretation will not get any faster if we only start translating after each sentence finishes, regardless of word order. Thus, we also need a method to choose when we can begin translation mid-sentence.

In our experiment (Section 5), we use the method of Fujita et al. [4] to decide the translation timing according to each phrase's right probability (RP). This method was designed for simultaneous speech translation, and decides in real time whether or not to start translating based on a threshold for each phrase's RP, which shows the degree to which the order of the source and target language can be expected to be the same. For phrases where the RP is high, it is unlikely that a reordering will occur, and thus we can start translation, even mid-sentence, with a relatively low chance of damaging the final output. On the other hand, if an RP is low, starting translation of the phrase prematurely may cause un-natural word ordering in the output. Thus, Fujita et al. choose a threshold for the RP of each phrase, and when the current phrase at the end of the input has an RP that exceeds the threshold, translation is started, but when the current phrase is under the threshold, the system waits for more words before starting translation.

While Fujita et al. calculated their RPs from translation data, there is a possibility that interpreters will use less reordering than translators for many source language phrases. To take account of this, we simply make the RP table from translation data and simultaneous interpretation data. Using this method, we can hope that the system will be able to choose earlier timing to translate without a degradation in the translation accuracy. We calculate the RP from translation and interpretation data by simply concatenating the data before calculation.

5. Experiment

5.1. Data

In our experiment, the task is translating TED talks from English to Japanese. We use the translation and the interpretation data from TED as described in Section 2. As this data is still rather small to train a reasonably accurate machine translation system, we also use the EIJIRO dictionary and the accompanying example sentences[2] in our training data.

Table 3: The number of words in the data we used for learning translation model (TM), language model (LM), tuning (tune) and test set (test). The kinds of data are TED translation data (TED-T), TED simultaneous interpretation data (TED-I) and a dictionary with its corresponding example sentences (DICT)

	TED-T	TED-I	DICT
TM/LM (en)	1.57M	29.7k	13.2M
TM/LM (ja)	2.24M	33.9k	19.1M
tune (en)	12.9k	12.9k	—
tune (ja)	19.1k	16.1k	—
test (en)	—	11.5k	—
test (ja)	—	14.9k	—

The details of the corpus are shown in Table 3. As simultaneous interpretation data for both training and testing, we use the data from the S rank interpreter. This is because the S rank interpreter has the longest experience of the three simultaneous interpreters, and as shown empirically in Section 3, is able to translate significantly more content than the A rank interpreter. As it is necessary to create sentence alignments between the simultaneous interpretation data and TED subtitles, we use the Champollion toolkit [12] to create the alignments for the LM/TM training data, and manually align the sentences for the tuning and testing data.

5.2. Toolkit and evaluation method

As a machine translation engine, we use the Moses [13] phrase-based translation toolkit. The tokenization script in the Moses toolkit is used as an English tokenizer. KyTea [14] is used as a Japanese tokenizer. GIZA++ [15] is used for word alignment and SRILM [16] is used to train a Kneser-Ney smoothed 5-gram LM. Minimum Error Rate Training [17] is used for tuning to optimize BLEU. The distortion limit during decoding is set to 12, which gave the best accuracy on the development set.

The system is evaluated by the translation accuracy and the delay. BLEU [9] and RIBES [10] are used to calculate translation accuracy. RIBES is an evaluation method that focuses on word reordering information, and is known to work well for the language pairs that have very different grammatical structure like English-Japanese. The delay D is calculated as $D = U + T$. U is the average amount of time that we must wait before we can start translating, and T is the time required for MT decoding. Note that, in this experiment, we make the simplifying assumption that we have 100% accurate ASR that can recognize each word in exactly real time,

[2] Available from http://eijiro.jp

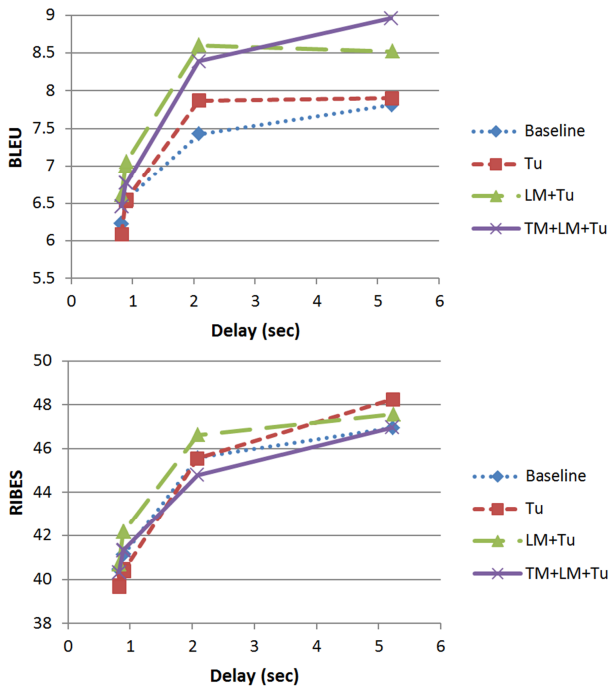


Figure 4: Result of machine translation system

and do not consider the time required for speech synthesis.

5.3. Result: Learning of the MT system

Simultaneous interpretation data is used in the three processes described in Section 4.1. To compare each variety of training, we experiment with 4 patterns:

Baseline: only translation data (w/o TED simultaneous interpretation data)

Tu: TED simultaneous interpretation data for tuning

LM+Tu: TED simultaneous interpretation data for LM training and tuning

TM+LM+Tu: TED simultaneous interpretation data for TM training, LM training and tuning

We decide the timing for translation according to the method described in Section 4.2, using a RP threshold of 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0.

The result of BLEU and delay is shown in the upper part of Figure 4. From these results, we can see that Tu does not show a significant improvement compared to the baseline, while LM+Tu and TM+LM+Tu show a significant improvement. For example, when the BLEU is 7.81[3], the delay is 5.23 seconds in the baseline, while in TM+LM+Tu the BLEU is 8.39, the delay is only 2.08 seconds. On the other hand, the result of RIBES and delay is shown in the lower part of Figure 4. In terms of RIBES, Tu, LM+Tu, and TM+LM+Tu do not show a significant improvement compared to the baseline. One of the reasons for this is tuning. When tuning, the parameters are optimized for BLEU, not RIBES. It should be noted that these numbers are all calculated using the S Rank interpreter's translations as a reference. In contrast, when we use the TED subtitles as a reference, the results for the baseline (BLEU=12.79, RIBES=55.36) were higher than those for TM+LM+Tu (BLEU=10.38, RIBES=53.94). From this experiment, we can see that by introducing simultaneous interpretation data in the training process of our machine translation system, we are able to create a system that produces output closer to that of a skilled simultaneous interpreter, although this may result in output that is further from that of time-unconstrained translators.

An example of results for the simultaneous interpreter, baseline, and TM+LM+Tu is shown in Table 4. From this example, we can see that the length of TM+LM+Tu is shorter than the baseline and is similar to the reference of simultaneous interpretation, as the length is adjusted during tuning. In this case, the reason for this is because the starting phrase in the baseline "見てみると" ("looking at") in baseline changes "では" ("ok") in TM+LM+Tu. Both translations are reasonable in this context, but the adapted system is able to choose the shorter one to reduce the number of words slightly. Another good example of how lexical choice was affected by adaptation to the simultaneous translations is the use of connectives between utterances. For example, the S rank simultaneous interpreter often connected two sentences by starting a sentence with the word "で" ("and"), likely to avoid long empty pauses while he was waiting for input. This was observed in 149 sentences out of 590 in the test set (over 25%). Our system was able to learn this distinct feature of simultaneous interpretation to some extent. In the baseline there were only 34 sentences starting with this word, while in TM+LM+Tu there were 81.

5.4. Result: Learning of translation timing

Next, we compare when the translation and the simultaneous interpretation data are used for learning of the RP (With TED-I) with when only translation data is used (W/O TED-I). The MT system is TM+LM+Tu for both settings.

The result is shown in Figure 5. From these two graphs, there is no difference in the translation accuracy and delay. We can hypothesize two reasons for this. First, the size of the simultaneous interpretation corpus is too small. The number of English words in the TED translation data is 1.57M, however, that in the TED simultaneous interpretation data is 29.7k. The second reason lies in the method we adopted for learning the RP table. In this experiment, the RP table is simply made by concatenating the translation data and simultaneous interpretation data. One potential way of solving this

[3]We speculate that the reason for these relatively low BLEU scores is the different grammatical structure between English and Japanese, and the highly stylized format of TED talks. Due to these factors, there is a lot of flexibility in choosing a translation, so the difference in lexical choice by translators might negatively affect the BLEU score.

Table 4: Example of translation results

	Sentence
Source	if you look at in the context of history you can see what this is doing
S Rank Reference	過去から / 流れを見てみますと / 災害は / このように / 増えています from the past / look at the context and / disasters are / like this / increasing
Baseline (RP 1.0)	見てみると / 歴史の中で / 見ることができます / これがやっていること looking at / in the history / you can see / what this is doing
TM+LM+Tu (RP 1.0)	では / 歴史の中で / 見ることができます / これがやっていること ok / in the history / you can see / what this is doing

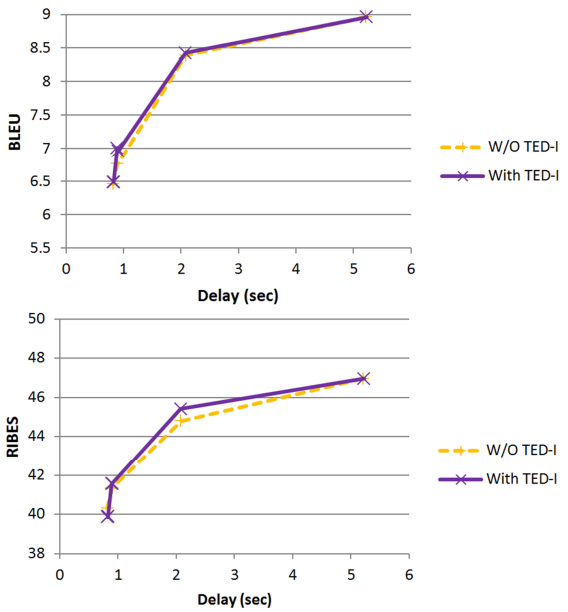


Figure 5: Result of dividing position

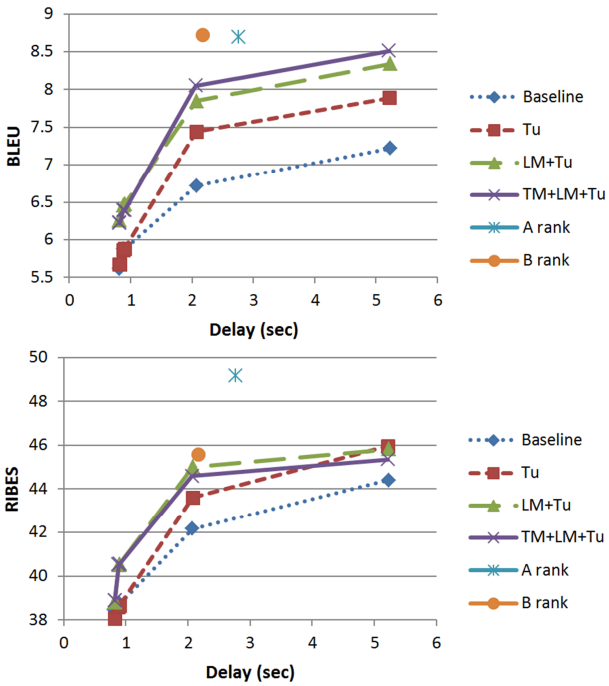


Figure 6: Result of comparing the system with human simultaneous interpreters

problem is, like we did for the TM, creating the table using the fill-up method.

5.5. Result: Comparing the system with human simultaneous interpreters

Finally, we compare the simultaneous ST system with human simultaneous interpreters. Simultaneous interpretation (and particularly that of material like TED talks) is a difficult task for humans, so it would be interesting to see how close are automatic systems are to achieving accuracy in comparison to imperfect humans. In the previous experiments, we assumed an ASR system that made no transcription errors, but if we are to compare with actual interpreters, this is an unfair comparison, as interpreters are also required to accurately listen to the speech before they translate. Thus, in this experiment, we use ASR results as input to the translation system. The word error rate is 19.36%. We show the results of our translation systems, as well as the A rank (4 years) and B rank (1 year) interpreters in Figure 6.

First, comparing the results of the automatic systems with Figure 4, we can see that the accuracy is slightly lower in terms of BLEU and RIBES. However the overall trend is almost same. From the view of BLEU, the system achieves results slightly lower than those of human simultaneous interpreters. However from the view of RIBES, the automatic system and B rank interpreter achieve similar results. So the performance of the system is similar, but likely slightly inferior to the B rank interpreter. It is also interesting to note the delay of the simultaneous interpreters. Around two seconds of delay is the shortest delay with which the system can translate while maintaining the translation quality. As well, the simultaneous interpreters begin to interpret two to three seconds after the utterance starts. We hypothesize that it is difficult to begin earlier than this timing while maintaining

the translation quality, both for humans and machines.

6. Conclusions

In this paper, we investigated the effects of constructing simultaneous ST system using simultaneous interpretation data for learning. As a result, we find the translation system grows closer to the translation style of a highly experienced professional interpreter. We also find that the translation accuracy has approached that of a simultaneous interpreter with 1 year of experience according to automatic evaluation measures. In the future, we are planning to do subjective evaluation, and analyze the differences in the style of translation between the systems in more detail.

7. Acknowledgments

Part of this work was supported by JSPS KAKENHI Grant Number 24240032.

8. References

[1] Roderick Jones. *Conference Interpreting Explained (Translation Practices Explained)*. St. Jerome Publishing, 2002.

[2] Koichiro Ryu, Atsushi Mizuno, Shigeki Matsubara, and Yasuyoshi Inagaki. Incremental Japanese spoken language generation in simultaneous machine interpretation. In *Proc. Asian Symposium on Natural Language Processing to Overcome language Barriers*, 2004.

[3] Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan Ladan Golipour, and Aura Jimenez. Real-time incremental speech-to-speech translation of dialogs. In *Proc. NAACL*, 2012.

[4] Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Simple, lexicalized choice of translation timing for simultaneous speech translation. In *Proc. 14th InterSpeech*, 2013.

[5] Matthias Paulik and Alex Waibel. Automatic translation from parallel speech: Simultaneous interpretation as mt training data. In *Proc. ASRU*, pages 496–501. IEEE, 2009.

[6] Vivek Kumar Rangarajan Sridhar, John Chen, and Srinivas Bangalore. Corpus analysis of simultaneous interpretation data for improving real time speech translation. In *Proceedings of InterSpeech*, 2013.

[7] Hitomi Toyama, Shigeki Matsubara, Koichiro Ryu, Nobuo Kawaguchi, and Yasuyoshi Inagaki. Ciair simultaneous interpretation corpus. In *Proc. Oriental COCOSDA*, 2004.

[8] Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Collection of a simultaneous translation corpus for comparative analysis (in submission). In *Proc. LREC 2014*, 2014.

[9] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318, Philadelphia, USA, 2002.

[10] Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. Automatic evaluation of translation quality for distant language pairs. In *Proc. EMNLP*, pages 944–952, 2010.

[11] Arianna Bisazza, Nick Ruiz, and Marcello Federico. Fill-up versus interpolation methods for phrase-based smt adaptation. In *Proc. IWSLT*, pages 136–143, 2011.

[12] Xiaoyi Ma. Champollion: A robust parallel text sentence aligner. In *Proc. LREC*, 2006.

[13] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL*, pages 177–180, Prague, Czech Republic, 2007.

[14] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proc. ACL*, pages 529–533, Portland, USA, June 2011.

[15] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[16] Andreas Stolcke. SRILM - an extensible language modeling toolkit. In *Proc. 7th International Conference on Speech and Language Processing (ICSLP)*, 2002.

[17] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proc. ACL*, 2003.

Improving the Minimum Bayes' Risk Combination of Machine Translation Systems

Jesús González-Rubio, Francisco Casacuberta

Departamento de sistemas informáticos y computación
Universitat Politècnica de València, camino de Vera s/n, 46022 Valencia, Spain
{jegonzalez,fcn}@dsic.upv.es

Abstract

We investigate the problem of combining the outputs of different translation systems into a minimum Bayes' risk consensus translation. We explore different risk formulations based on the BLEU score, and provide a dynamic programming decoding algorithm for each of them. In our experiments, these algorithms generated consensus translations with better risk, and more efficiently, than previous proposals.

1. Introduction

Machine translation (MT) is a fundamental technology and a core component of language processing systems. However, MT systems are still far from perfect [1]. The combination of multiple MT systems is a promising research direction to improve the quality of current MT technology. The key idea of system combination [2] is that it is often very difficult to find the real best system for the task at hand, while different systems can exhibit complementary strengths and limitations. Thus, a proper combination of systems could be more effective than using a single monolithic system.

A simple, yet effective, system combination method for MT was proposed by González-Rubio et al., [3]. The authors describe minimum Bayes' risk system combination (MBRSC), a method to combine the outputs of multiple MT systems into a consensus translation with maximum expected BLEU [4] score. Previous combination methods either implement sophisticated decision functions to select one of the provided translations [5, 6, 7], or generate new consensus translations by combining the best subsequences of the provided translations by means of a Viterbi-like search on a confusion network [8, 9, 10]. MBRSC aims at gathering together the advantages of sentence-selection and subsequence-combination methods. In comparison to sentence-selection methods, MBRSC also implements a sophisticated minimum Bayes' risk (MBR) classifier, and additionally, it is able to generate new consensus translations that include the "best" subsequences from different individual translations. Regarding subsequence-combination methods, MBRSC can also generate new consensus translations different from the provided translations, and also, the final consensus translation has the best expected score with respect to the widespread BLEU score.

Despite these advantages, the original implementation of MBRSC [3] (§2) presented some flaws, e.g. the proposed gradient ascent decoding, that, in our opinion, prevents the method from revealing its full potential. Here, we propose new decoding algorithms for MBRSC based on the dynamic programming [11] (DP) paradigm. We study two different approaches to compute the BLEU-based risk. On the one hand, we instantiate DP decoding to use the original BLEU risk over expected counts (§3) so our results are comparable to those in [3]. In practice, this approach is implemented as a beam search [12]. On the other hand, we implement an actual exact DP decoding using the linear approximation to the BLEU score proposed in [13] to compute the risk (§4). Then, we provide an extensive empirical study (§5) of the proposed decoding algorithms in comparison to the original MBRSC proposal. Finally, we conclude with a summary of our contributions.

2. Minimum Bayes' Risk System Combination

2.1. MBRSC Model and Decision Function

We now describe the original MBRSC proposal in [3]. Given K MT systems, MBRSC models the probability of a sentence $\mathbf{y}$ to be a translation of a source sentence $\mathbf{x}$ as a weighted ensemble [14]:

$$P(\mathbf{y} \mid \mathbf{x}) = \sum_{k=1}^{K} \alpha_k \cdot P_k(\mathbf{y} \mid \mathbf{x}) \tag{1}$$

where $P_k(\mathbf{y} \mid \mathbf{x})$ denotes the probability distribution over translations modeled by system k. Free parameters $\{\alpha_1, \ldots, \alpha_K\}$ are scaling factors that denote the relative importance of each system ($\sum_{k=1}^{K} \alpha_k = 1$).

Given a loss function $\mathrm{L}(\mathbf{y}, \mathbf{y}')$ between a candidate translation $\mathbf{y}$ and a reference translation $\mathbf{y}'$, the optimal decision function for the ensemble model of MBRSC is an instance of the MBR classifier [15]:

$$\begin{aligned}\hat{\mathbf{y}} &= \arg\min_{\mathbf{y}\in\mathcal{Y}} \mathrm{R}(\mathbf{y} \mid \mathbf{x}) \\ &= \arg\min_{\mathbf{y}\in\mathcal{Y}} \mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}\left[\mathrm{L}(\mathbf{y}, \mathbf{y}')\right] \\ &= \arg\min_{\mathbf{y}\in\mathcal{Y}} \sum_{\mathbf{y}'\in\mathcal{Y}} P(\mathbf{y}' \mid \mathbf{x}) \cdot \mathrm{L}(\mathbf{y}, \mathbf{y}') \end{aligned} \quad (2)$$

where $\mathrm{R}(\mathbf{y} \mid \mathbf{x})$ denotes the Bayes' risk, namely the expected loss ($\mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}\left[\mathrm{L}(\mathbf{y}, \mathbf{y}')\right]$), of translation $\mathbf{y}$, and $\mathcal{Y}$ denotes the whole target language.

MBRSC uses the widespread BLEU [4] metric as loss function. The BLEU score $\mathrm{B}(\mathbf{y}, \mathbf{y}')$ between a candidate translation $\mathbf{y}$ and a reference $\mathbf{y}'$ is given by:

$$\mathrm{B}(\mathbf{y}, \mathbf{y}') = \left(\prod_{n=1}^{4} \rho_n(\mathbf{y}, \mathbf{y}')\right)^{\frac{1}{4}} \cdot \phi(\mathbf{y}, \mathbf{y}') \quad (3)$$

where $\rho_n(\mathbf{y}, \mathbf{y}')$ is the precision of n-grams of size n between $\mathbf{y}$ and $\mathbf{y}'$, and $\phi(\mathbf{y}, \mathbf{y}')$ is a brevity penalty, that penalizes short translations:

$$\rho_n(\mathbf{y}, \mathbf{y}') = \frac{\sum_{\mathbf{w}\in\mathcal{W}_n(\mathbf{y})} \min(\#_{\mathbf{w}}(\mathbf{y}), \#_{\mathbf{w}}(\mathbf{y}'))}{\sum_{\mathbf{w}\in\mathcal{W}_n(\mathbf{y})} \#_{\mathbf{w}}(\mathbf{y})} \quad (4)$$

$$\phi(\mathbf{y}, \mathbf{y}') = \min\left(\exp\left(1 - \frac{|\,\mathbf{y}'\,|}{|\,\mathbf{y}\,|}\right), 1\right) \quad (5)$$

where $\mathcal{W}_n(\mathbf{y})$ is the set of n-grams of size n in $\mathbf{y}$, $\#_{\mathbf{w}}(\mathbf{y})$ is the count of n-gram $\mathbf{w}$ in $\mathbf{y}$, and $|\,\mathbf{y}\,|$ denotes the length of translation $\mathbf{y}$.

BLEU is a percentage with a value of one denoting an exact match between $\mathbf{y}$ and $\mathbf{y}'$. Thus, we rewrite the MBRSC decision function in Equation (2) substituting the $\arg\min_{\mathbf{y}\in\mathcal{Y}}$ operator by an $\arg\max_{\mathbf{y}\in\mathcal{Y}}$:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} \sum_{k=1}^{K} \alpha_k \cdot \underbrace{\left(\sum_{\mathbf{y}'\in\mathcal{Y}} P_k(\mathbf{y}' \mid \mathbf{x}) \cdot \mathrm{B}(\mathbf{y}, \mathbf{y}')\right)}_{\text{system-specific loss}} \quad (6)$$

This formulation assumes that all systems share the same domain of translations ($\mathcal{Y}$) which in practice it is not always true. In practice, MBRSC takes as input a representation, e.g. an N-best list, of the candidate translations of each system and assumes that any other translation not in the provided representation has zero probability of being generated by that system.

Optimum values for scaling factors α_k are estimated by minimum error rate training [16] optimizing BLEU on a separate development set.

2.2. MBRSC Decoding

The direct implementation of Equation (6) has a high temporal complexity in $O(|\,\mathcal{Y}\,|^2 \cdot I)$, where $|\,\mathcal{Y}\,|$ denotes the number of candidate translations, and I represents the maximum translation length given that $\mathrm{B}(\mathbf{y}, \mathbf{y}')$ can be computed in $O(\max(|\,\mathbf{y}\,|, |\,\mathbf{y}'\,|))$ time. Since the number of candidate translations may be quite large, an exhaustive enumeration of all of them is often unfeasible. González-Rubio et. al [3] address this challenge by dividing Equation (6) into two sub-problems: the computation of the risk, namely the expected BLEU score, of each translation, and the actual search for the optimal consensus translation ($\arg\max_{\mathbf{y}\in\mathcal{Y}}$).

Given that BLEU references the reference translation $\mathbf{y}'$ only via its n-gram counts (see Equation (3)), MBRSC follows [17] to formalize an efficient alternative to the exact risk in Equation (6). Instead of computing the expected BLEU score of translation $\mathbf{y}$, MBRSC computes the BLEU score of $\mathbf{y}$ with respect to the expected n-gram counts $\mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\#_{\mathbf{w}}(\mathbf{y}')]$ in the alternative candidate translations of $\mathbf{x}$:

$$\begin{aligned}\mathrm{R}(\mathbf{y} \mid \mathbf{x}) &= \mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\mathrm{B}(\mathbf{y}, \mathbf{y}')] \\ &\approx \widetilde{\mathrm{B}}(\mathbf{y}, \mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\#_{\mathbf{w}}(\mathbf{y}')]) \\ &= \left(\prod_{n=1}^{4} \widetilde{\rho_n}(\mathbf{y}, \mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\#_{\mathbf{w}}(\mathbf{y}')])\right)^{\frac{1}{4}} \cdot \\ &\quad \widetilde{\phi}(\mathbf{y}, \mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\#_{\mathbf{w}}(\mathbf{y}')]) \end{aligned} \quad (7)$$

where $P(\mathbf{y}' \mid \mathbf{x})$ is the ensemble probability in Equation (1), and $\rho_n(\mathbf{y}, \mathbf{y}')$ and $\phi(\mathbf{y}, \mathbf{y}')$ are reformulated as functions of expected n-gram counts.

Regarding the actual search, MBRSC implements a two-step algorithm. First, it performs a conventional MBR sentence-selection decoding [18] to obtain an initial consensus translation. Then, a gradient ascent algorithm refines that initial solution by the iterative application of different edit operations (substitution, insertion, and deletion of single words) searching for an improvement in risk. Algorithm 1 depicts this gradient ascent decoding algorithm. Since the risk ($\mathrm{R}(\mathbf{y} \mid \mathbf{x})$ in Equation (7)) can be computed in $O(I)$[1], the com-

[1] Expected n-gram counts can be computed in advance.

Algorithm 1: MBRSC gradient ascent search [3].

input : $\mathbf{y}_0$ (initial solution)
Σ (target language vocabulary)
I (maximum translation length)
output : $\hat{\mathbf{y}}, \mathrm{R}(\hat{\mathbf{y}} \mid \mathbf{x})$ (best translation and its score)
auxiliary : $\mathrm{R}(\mathbf{y} \mid \mathbf{x})$ (expected BLEU score of $\mathbf{y}$)
$\mathrm{sub}(\mathbf{y}, y, i)$ (replaces i^{th} word of $\mathbf{y}$ by y)
$\mathrm{del}(\mathbf{y}, i)$ (deletes the i^{th} word of $\mathbf{y}$)
$\mathrm{ins}(\mathbf{y}, y, i)$ (inserts y as the i^{th} word of $\mathbf{y}$)

begin
  $\hat{\mathbf{y}} \leftarrow \mathbf{y}_0$;
  repeat
    $\mathbf{y}_c \leftarrow \hat{\mathbf{y}}$;
    for $1 \leq i \leq |\,\mathbf{y}_c\,|$ **do**
      $\hat{\mathbf{y}}_s \leftarrow \mathbf{y}_c$; $\hat{\mathbf{y}}_i \leftarrow \mathbf{y}_c$;
      for $y \in \Sigma$ **do**
        $\mathbf{y}_s \leftarrow \mathrm{sub}(\mathbf{y}_c, y, i)$;
        if $\mathrm{R}(\mathbf{y}_s \mid \mathbf{x}) \geq \mathrm{R}(\hat{\mathbf{y}}_s \mid \mathbf{x})$ **then**
          $\hat{\mathbf{y}}_s \leftarrow \mathbf{y}_s$;
        $\mathbf{y}_i \leftarrow \mathrm{ins}(\mathbf{y}_c, y, i)$;
        if $\mathrm{R}(\mathbf{y}_i \mid \mathbf{x}) \geq \mathrm{R}(\hat{\mathbf{y}}_i \mid \mathbf{x})$ **then**
          $\hat{\mathbf{y}}_i \leftarrow \mathbf{y}_i$;
      $\hat{\mathbf{y}}_d \leftarrow \mathrm{del}(\mathbf{y}_c, i)$;
      $\hat{\mathbf{y}} \leftarrow \arg\max_{\mathbf{y}' \in \{\hat{\mathbf{y}}, \hat{\mathbf{y}}_s, \hat{\mathbf{y}}_i, \hat{\mathbf{y}}_d\}} \mathrm{R}(\mathbf{y}' \mid \mathbf{x})$
  until $(\mathrm{R}(\hat{\mathbf{y}} \mid \mathbf{x}) \leq \mathrm{R}(\mathbf{y}_c \mid \mathbf{x}))$ || $(\hat{\mathbf{y}} \geq I)$;
  return $\hat{\mathbf{y}}, \mathrm{R}(\hat{\mathbf{y}} \mid \mathbf{x})$;
end

plexity of the main loop is $O(I^2 \cdot |\Sigma|)$, and usually only a moderate number of iterations (< 10) are needed to converge. Hence, the complete two-step decoding has a complexity in $O(N^2 + I^2 \cdot |\Sigma|)$, where N is the number of translations under consideration in the preliminary sentence-selection decoding.

3. MBRSC Dynamic Programming Decoding

The main drawback of the originally proposed gradient ascent decoding is that it is sensitive to an initial solution which makes it prone to get stuck in local optima. Next, we propose a more sophisticated approach by formalizing MBRSC decoding as a DP problem.

Under the DP framework, decoding is interpreted as a sequence of decisions that incrementally generate new translation hypotheses. Starting with an empty hypothesis, hypotheses of size i are expanded with one more target word $y \in \Sigma$ to create new hypotheses of size $i+1$. This search space can be represented as a directed acyclic graph where the states denote partial hypotheses and the edges are labeled with expansion words.

Among all possible translations, we are interested in that of the higher expected BLEU score. In this case, since two hypotheses sharing the same n-gram counts are indistinguishable, each state of the graph can be represented by a specific bag (namely a specific multiset) $\mathcal{N}$ of n-grams. We define $Q(\mathcal{N}, \mathbf{y}) = q$ where q is the maximum score of a path leading from the initial state to the state $(\mathcal{N})$, and $\mathbf{y}$ is the corresponding translation hypothesis. We also define $\widehat{Q} = \hat{q}$ as the final state of the optimal translation $\hat{\mathbf{y}}$. Finally, the following DP recursion equations allow us to retrieve the path of maximum score in such a search graph:

$$Q(\emptyset, \text{""}) = 0$$

$$Q(\mathcal{N}_e, \mathbf{y}_e) = \max_{\substack{y \in \Sigma \cup \{\$\}: \\ \forall (\mathcal{N}_p, \mathbf{y}_p),\ \mathbf{y}_e = \mathbf{y}_p\, y \\ \mathcal{N}_e = \mathcal{N}_p \cup \Theta(\mathbf{y}_p, y)}} \widetilde{\mathrm{B}}(\mathbf{y}_e, \mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\#_{\mathbf{w}}(\mathbf{y}')])$$

$$\widehat{Q} = \max_{\substack{\forall (\mathcal{N}_p, \mathbf{y}_p) \\ \hat{\mathbf{y}} = \mathbf{y}_p\, \$}} \widetilde{\mathrm{B}}(\hat{\mathbf{y}}, \mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\#_{\mathbf{w}}(\mathbf{y}')])$$

where the end-of-sentence symbol, \$, denotes a complete translation, and function $\Theta(\mathbf{y}_p, y)$ returns the new n-grams generated when expanding hypothesis $\mathbf{y}_p$ with word y. For example, given the hypothesis $\mathbf{y}_p$="we are faced with" and the expansion word y="enormous", the expanded hypothesis $\mathbf{y}_e$="we are faced with enormous" contains four[2] n-grams more than $\mathbf{y}_p$: "enormous", "with enormous", "faced with enormous", and "are faced with enormous".

In the DP recursion equations, all target language words are considered as potential expansion options for every hypothesis. However, not all word sequences form correct natural language sentences. E.g., given the example above, it is clear that word y="enormous" can be a valid expansion option while word y="with" cannot. Thus, we consider $y \in \Sigma \cup \{\$\}$ as a valid expansion word for hypothesis $\mathbf{y}_p$ only if at least one of the new n-grams ($\mathbf{w} \in \Theta(\mathbf{y}_p, y)$) in the resulting expanded hypothesis $\mathbf{y}_e = \mathbf{y}_p\, y$ has an expected count above zero:

$$\Delta(\mathbf{y}_p) = \{y \mid \exists\, \mathbf{w} \in \Theta(\mathbf{y}_p, y)\ \wedge \mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\#_{\mathbf{w}}(\mathbf{y}')] > 0\}$$

Unfortunately, due to the exponential number of states[3], we cannot expect to efficiently implement the recursion equations above. In practice, we use a beam search algorithm [12] with pruning. Specifically, for each size i, we keep only the M best-scoring hypotheses and discard the rest of them. To assure a fair competition between hypotheses, the score of each of them

[2] BLEU considers n-grams up to size four.

[3] The number is exponential in the size of the vocabulary [19].

Algorithm 2: Beam search for MBRSC.

input : $\mathbf{x}$ (source language sentence),
M (pruning parameter),
I (maximum translation length)
output : $\hat{y}, \hat{q}$ (optimal translation and its score)
auxiliary : $\Theta(\mathbf{y}, y)$ (new n-grams after expanding hypothesis $\mathbf{y}$ with word y),
$\Delta(\mathbf{y})$ (expansion words for hypothesis $\mathbf{y}$),
$\overline{\mathrm{R}}(\mathbf{y} \mid \mathbf{x})$ (complete score of $\mathbf{y}$),
$\Pi(i, N)$ (non-pruned states of size i)

```
1  begin
2    Q(∅, "") ← 0;  ŷ ← "";  Q̂ ← 0;
3    for i = 0 to I do
4      forall (N_p, y_p) ∈ Π(i, N) do
5        forall y ∈ Δ(y_p) do
6          y_e ← y_p y;  q_e ← R̄(y_e | x);
7          if y == $ then
8            q̂ ← Q̂;
9            if q_e > q̂ then
10             ŷ ← y_e;  Q̂ ← q_e;
11         else
12           N_e ← N_p ⋃ Θ(y_p, y);
13           q ← Q(N_e, ·);
14           if q_e > q then
15             Q(N_e, y_e) ← q_e;
16   return ŷ, Q̂;
17 end
```

is given by a combination of its score so far, and an estimate of the rest score to complete the translation. Similarly as done in [20], we perform a light decoding process (considering at each step only the single best expansion) to estimate the complete translation that can be obtained from each hypothesis. The score of these complete translations are then used as the complete scores $\overline{\mathrm{R}}(\mathbf{y} \mid \mathbf{x})$ of the partial hypotheses.

Algorithm 2 shows the proposed beam search algorithm with pruning. It takes as input a source sentence $\mathbf{x}$, the number of hypotheses to keep after pruning (M), and the maximum translation length under consideration (I). We use some auxiliary functions: $\Theta(\mathbf{y}, y)$ returns the set of new n-grams generated in the expansion of hypothesis $\mathbf{y}$ with word y, $\Delta(\mathbf{y})$ returns the valid expansion words for $\mathbf{y}$, $\overline{\mathrm{R}}(\mathbf{y} \mid \mathbf{x})$ returns the complete score of $\mathbf{y}$, and $\Pi(i, M)$ denotes the M best states of size i; lower-scoring states are pruned out.

To avoid repeated computations, the first loop in Algorithm 2 performs a breadth-first exploration of the search graph. Additionally, this loop introduces an upper bound to the maximum translation size under consideration, and thus, to the number of iterations of the algorithm. At each iteration, line 4 loops over the non-pruned states that remain from the previous iteration. For each of these predecessor states, line 5 loops over the corresponding expansion words. Given a predecessor state $(\mathcal{N}_p, \mathbf{y}_p)$ and a valid expansion word y, we compute the complete score q_e of the expanded hypothesis $\mathbf{y}_e = \mathbf{y}_p\, y$ (line 6). If the expanded hypothesis is a complete translation ($y == \$$) and it improves the score $\widehat{Q}$ of the current best consensus translation, we then update it (lines 7–10). If not, we first compute the bag of n-grams $\mathcal{N}_e$ of the expanded hypothesis (line 12). Then, if the score q_e of the expanded hypothesis improves the score stored in the corresponding successor state $(\mathcal{N}_e, \cdot)$ (line 14), we update the state.

The proposed beam search algorithm with pruning has a computational complexity in $O(I^2 \cdot M \cdot D)$, where M denotes the pruning parameter that controls the number of predecessor states in line 4, D denotes the maximum number of expansion words in line 5, and I is the maximum translation size in line 3. The extra $O(I)$ factor is given by the score computation in line 6.

4. MBRSC DP Search for Linear BLEU

A potential drawback of decoding Algorithm 2 is that it cannot exploit the full potential of the DP framework. The problem stems in the BLEU based risk proposed in [3]: the n-gram count clippings in its formulation, see Equation (4), make impossible to compute it incrementally. To address this problem, we import the linear approximation to the logarithm of the BLEU scores proposed in [13]:

$$\log(\mathrm{B}(\mathbf{y}, \mathbf{y}')) \approx \lambda_0 |\, \mathbf{y} \,| + \sum_{\mathbf{w} \in \mathcal{W}(\mathbf{y})} \lambda_{\mathbf{w}} \#_{\mathbf{w}}(\mathbf{y}) \delta_{\mathbf{w}}(\mathbf{y}') \qquad (8)$$

where $\mathcal{W}(\mathbf{y})$ is the complete set of n-grams (up to size four) in $\mathbf{y}$, λ_0 and $\lambda_{\mathbf{w}}$ are free parameters, and $\delta_{\mathbf{w}}(\mathbf{y}')$ is an indicator feature whose value is equal to one if n-gram $\mathbf{w}$ is present in $\mathbf{y}'$ and zero otherwise. Given this BLEU approximation, the risk of a candidate translation $\mathbf{y}$ is given by:

$$\mathrm{R}(\mathbf{y} \mid \mathbf{x}) = \lambda_0 |\, \mathbf{y} \,| + \sum_{\mathbf{w} \in \mathcal{W}(\mathbf{y})} \lambda_{\mathbf{w}} \#_{\mathbf{w}}(\mathbf{y}) \mathbb{E}_{P(\mathbf{y}' \mid \mathbf{x})}[\delta_{\mathbf{w}}(\mathbf{y}')] \qquad (9)$$

where $\mathbb{E}_{P(\mathbf{y}' \mid \mathbf{x})}[\delta_{\mathbf{w}}(\mathbf{y}')]$ denotes the expected probability of n-gram $\mathbf{w}$ to be present. Values $\lambda_0, \lambda_{\mathbf{w}}$ can be computed from the n-gram precision statistics of a separate development set [13]. Gradient ascent decoding

can also implement this risk formulation by using Equation (9) as risk function $\mathrm{R}(\mathbf{y} \mid \mathbf{x})$ in Algorithm 1.

Note that the BLEU risk over expected counts in Equation (7) yields a decoding alternative to MBR using BLEU, while the linear BLEU risk in Equation (9) results in a MBR decoding for an alternative to BLEU.

Using the linear BLEU risk in Equation (9), two partial hypotheses that share their last three words are indistinguishable. Hence, the states in the corresponding DP search graph can be represented by a particular three-word history $\boldsymbol{\sigma}$. To distinguish between hypotheses of different size, we also index the search states by the size of the best hypothesis that arrives to the state. We define $Q(i, \boldsymbol{\sigma})$ as the maximum score of a path leading from the initial state to the state $(i, \boldsymbol{\sigma})$, and $\widehat{Q}$ as the score of the optimal translation $\hat{\mathbf{y}}$. Finally, we obtain the following DP recursion equations:

$$Q(0, "") = 0$$

$$Q(i, \boldsymbol{\sigma}_e) = \max_{\substack{y \in \Sigma: \\ q_p = Q(i-1, \boldsymbol{\sigma}_p) \\ \mathbf{y}_e = \boldsymbol{\sigma}_p\, y \\ \boldsymbol{\sigma}_e = \text{tail}(\boldsymbol{\sigma}_p\, y)}} q_p + \lambda_0 + \sum_{\mathbf{w} \in \Theta(\boldsymbol{\sigma}_p, y)} \lambda_{\mathbf{w}} \mathbb{E}_{P(\mathbf{y}' \mid \mathbf{x})}[\delta_{\mathbf{w}}(\mathbf{y}')]$$

$$\widehat{Q} = \max_{\substack{q_p = Q(\cdot, \boldsymbol{\sigma}_p) \\ \boldsymbol{\sigma}_e = \text{tail}(\boldsymbol{\sigma}_p\, \$)}} q_p + \lambda_0 + \sum_{\mathbf{w} \in \Theta(\boldsymbol{\sigma}_p, \$)} \lambda_{\mathbf{w}} \mathbb{E}_{P(\mathbf{y}' \mid \mathbf{x})}[\delta_{\mathbf{w}}(\mathbf{y}')]$$

where $\text{tail}(\boldsymbol{\sigma}\, y)$ returns the last three words of word sequence $\boldsymbol{\sigma}\, y$, and $\Theta(\boldsymbol{\sigma}, y)$ returns the new n-grams generated when extending history $\boldsymbol{\sigma}$ with word y.

Since the number of states is at most cubical with the target vocabulary, these recursive equations can be implemented exactly. Algorithm 3 depicts DP decoding using linear BLEU risk. It takes as input the indicator feature expectations ($\mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\delta_{\mathbf{w}}(\mathbf{y}')]$), the values for the free parameters of linear BLEU ($\lambda_0, \lambda_{\mathbf{w}}$), and the maximum translation length under consideration (I). At each iteration the algorithm loops over the predecessor states (line 4) and the corresponding expansion words (line 5). Given a predecessor state $(i, \boldsymbol{\sigma}_p)$, we compute the score q_e of the expanded hypothesis (line 6), and if q_e improves the score in the corresponding successor state $(i+1, \boldsymbol{\sigma}_e)$ (line 8), we update it and the corresponding backpointer $B(i+1, \boldsymbol{\sigma}_e)$. Finally, backpointer variables allow us to retrieve the highest-scoring consensus translation.

This DP algorithm has a computational complexity in $O(I \cdot |\Sigma|^3 \cdot D)$, where I is the maximum translation length in line 3, $|\Sigma|$ denotes the size of the target vocabulary that controls the number of predecessor states in line 4, and D denotes the maximum number of expansion words in line 5.

Algorithm 3: MBRSC DP search for linear BLEU.

input : $\mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\delta_{\mathbf{w}}(\mathbf{y}')]$ (indicator feature expectations),
$\lambda_0, \lambda_{\mathbf{w}}$ (free parameters of linear BLEU),
I (maximum translation length)
output : $Q(\cdot, \cdot)$ (search graph),
$B(\cdot, \cdot)$ (backpointer variables)
auxiliary : $\text{tail}(\mathbf{y})$ (returns the last three words of $\mathbf{y}$),
$\Theta(\mathbf{y}, y)$ (new n-grams after expanding hypothesis $\mathbf{y}$ with word y),
$\Delta(\mathbf{y})$ (set of expansion words for $\mathbf{y}$)

1 **begin**
2 $\quad Q(\cdot, \cdot) \leftarrow 0$;
3 $\quad$ **for** $i = 0$ **to** I **do**
4 $\quad\quad$ **forall** $\boldsymbol{\sigma}_p \in Q(i, \cdot)$ **do**
5 $\quad\quad\quad$ **forall** $y \in \Delta(\boldsymbol{\sigma}_p)$ **do**
6 $\quad\quad\quad\quad q_e \leftarrow Q(i, \boldsymbol{\sigma}_p) + \lambda_0 + \sum_{\mathbf{w} \in \Theta(\boldsymbol{\sigma}_p, y)} \lambda_{\mathbf{w}} \mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\delta_{\mathbf{w}}(\mathbf{y}')]$;
7 $\quad\quad\quad\quad \boldsymbol{\sigma}_e \leftarrow \text{tail}(\boldsymbol{\sigma}_p\, y)$;
8 $\quad\quad\quad\quad$ **if** $q_e > Q(i+1, \boldsymbol{\sigma}_e)$ **then**
9 $\quad\quad\quad\quad\quad Q(i+1, \boldsymbol{\sigma}_e) \leftarrow q_e$;
10 $\quad\quad\quad\quad\quad B(i+1, \boldsymbol{\sigma}_e) \leftarrow (i, \boldsymbol{\sigma}_p)$;
11 **end**

5. Experiments

5.1. Experimental Setup

We now describe the experimentation carried out to evaluate the proposed decoding algorithms. Experiments were performed on the French–English corpus from the translation task of the 2009 workshop on statistical MT [21]. The corpus contains a development and a test partition with 502 and 2525 sentences respectively. We combined the outputs of the five MT systems that submitted lists of N-best translations. The next table displays the average number of translations for each source sentence, and BLEU scores for the single best translations of each system.

System	#avg_trans_N	BLEU [%]
A	13	24.8
B	9	25.2
C	41	25.8
D	263	25.8
E	126	26.4

Translations were tokenized and lower-cased before combination. We report case-insensitive results to factor out the effect of true-casing from the effect of computing the consensus translation.

The separate development set was used to compute the values of the parameters $(\lambda_0, \lambda_{\mathbf{w}})$ of linear BLEU. The maximum translation length I was always set equal to the length of the longest provided translation; a more sophisticated length model could be devised, but this is a research direction beyond the scope of this article. Except stated otherwise, all experiments were carried out using uniform ensemble weights (α_k in Equation (1)). This approach defines a controlled environment that assures a fair comparison between the different decoding algorithms. For each source sentence, we combined all the translation provided by the five individual systems, on average, about 450 translations. We used these translations to compute the expected n-gram counts $\mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\#_{\mathbf{w}}(\mathbf{y}')]$, and the n-grams expectations $\mathbb{E}_{P(\mathbf{y}'|\mathbf{x})}[\delta_{\mathbf{w}}(\mathbf{y}')]$ for each source sentence.

5.2. Assessment Measures

We present translation quality results in terms of BLEU [4] (see Equation (3)), and TER [22]. TER measures the number of words that must be edited[4] to convert the candidate translation into the reference translation. Since MBRSC is designed to optimize BLEU, we expect improvements in BLEU to be particularly important. TER scores are reported to independently assess BLEU results. We also measure the statistical significance of the results by bootstrap re-sampling [23].

5.3. Preliminary Experiments

We carried out a preliminary series of experiments to study how the number of hypotheses kept after pruning (M) affects the performance of Algorithm 2 in terms of translation quality and decoding time[5]. Figure 1 displays the quality of the generated consensus translations (on the left vertical axis) and the total decoding time (on the right vertical axis) as functions of M. We observed that decoding time increased linearly with M (note that M is log-scaled in Figure 1) while the quality of the consensus translations stayed approximately constant with slight improvements for larger M values.

Given these results, we considered that a value $M = 10$ provided the optimal trade-off between translation quality and decoding time. Thus, this is the value used in the following experiments.

[4] Valid edit operations are: deletion, insertion and substitution of single words, and shift of word sequences

[5] In a PC with an Intel Core© i5-3570K processor (3.40 GHz.).

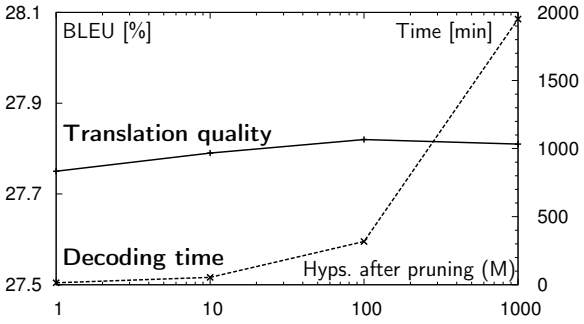


Figure 1: BLEU score (on the left vertical axis) and decoding time (on the right vertical axis) obtained by the beam search using BLEU risk on expected n-gram counts (Algorithm 2) as a function of the number of hypotheses kept after pruning (M).

5.4. Results

Table 1 displays BLEU and TER scores for the consensus translations generated by MBRSC using different decoding algorithms and risk formulations. We also report results for the best and worst single systems.

We first present results for sentence-selection decoding [18]. The risk of each candidate translation was computed by exhaustively calculating its BLEU-based risk with respect to the rest of the provided translations as in Equation (6). Results for both risk functions showed a substantial improvement over the best system: $\sim +0.9$ BLEU. Then, we used these sentence-selection consensus translations as initial solutions for the gradient ascent decoding proposed in [3] (Algorithm 1). Results for BLEU risk on expected n-gram counts slightly improved results for sentence-selection decoding: $+0.3$ BLEU and -0.1 TER. In contrast, results for linear BLEU risk showed an important degradation in performance: -0.9 BLEU and $+3.4$ TER. Finally, we generated consensus translations using BLEU risk over expected n-gram counts (Algorithm 2), and linear BLEU risk (Algorithm 3). Results for BLEU risk on expected counts slightly improved the results of the gradient ascent decoding: $+0.1$ BLEU and -0.3 TER. Regarding linear BLEU risk, it again exhibited the same poor performance observed for gradient ascent decoding.

Despite being scarce, the difference in translation quality between the proposed decoding algorithms and the original gradient ascent algorithm were statistically significant: 85% confidence for BLEU risk over expected counts, and 99% confidence for linear BLEU risk. Moreover, when we measured the risk scores of the generated consensus translations, we found that for

System setup		BLEU[%]	TER[%]
worst single system		24.8	60.4
best single system		26.4	56.0
Sentence-selection [18]	EC	27.4	55.5
	LB	27.2	56.2
Gradient ascent (Algorithm 1)	EC	27.7	55.4
	LB	26.3	59.6
BS (Algorithm 2)	EC	**27.8**	**55.1**
DP (Algorithm 3)	LB	26.8	57.8

Table 1: Quality of the consensus translations generated by different MBRSC setups. BS stands for beam search, EC for BLEU risk over expected counts (Equation (7)), and LB for linear BLEU risk (Equation (9)).

53% of the sentences DP-based search found a better-scoring output than gradient ascent decoding (47%).

We performed additional experiments where the values of the ensemble weights (α_k in Equation (1)) were trained to optimize BLEU in the development corpus. Results were similar to those in Table 1. For instance, beam search with risk over expected counts scored 28.1 BLEU while gradient ascent scored 27.8 BLEU. However, now DP-based search generated better-scoring consensus translation for 93% of the sentences. The scarce improvement with respect to the use of uniform values can be explained by the similar quality of the systems being combined, see §5.1.

We also compared DP search and gradient ascent search in terms of decoding time. We estimate decoding time by the number of times each algorithm calls the risk-computation function $\mathrm{R}(\mathbf{y} \mid \mathbf{x})$ during the generation of consensus translations for the whole corpus. We report this count instead of the actual decoding time to filter out the potential effects of the particular implementation of each algorithm. We observed that gradient ascent made ~ 23 millions calls to the risk function, while DP decoding made ~ 15 million calls including those involved in the estimation of the rest score. For instance, total decoding time for DP using BLEU risk over expected counts was about 55 minutes (~ 1.3 seconds per sentence).

Finally, we conclude that the proposed DP decoding is both more effective and efficient than the original gradient ascent decoding proposed in [3].

Regarding the low performance of linear BLEU risk, we consider that it was due to the the lack of n-gram count clippings in the linear BLEU risk formulation. Consensus translations obtained with linear BLEU risk tend to contain repeated instances of highly-probable n-grams which resulted in longer consensus translations (27.8 words on average) than the ones generated using BLEU risk over expected counts (26.4 words), and also longer than the average length (26.0 words) of the reference translations. Table 2 shows various examples of these erroneous consensus translations generated by Algorithm 3. Given the adequate performance of linear BLEU risk in our sentence-selection experiments and in previous works [13, 7], we conclude that linear BLEU is an effective loss function to be used in sentence-selection methods, but due to the lack of n-gram count clippings, it fails at scoring the new translations explored though decoding by subsequence-combination algorithms. The inclusion of more features, such as a language model, in the formulation of linear BLEU risk may mitigate this effect.

Alg. 2: we have made great progress . **Alg. 3:** we have made great progress . *we have made*
Alg. 2: it seems to be clear that it is better to buy only a phone . **Alg. 3:** *to be clear that* it seems to be clear that it is better to buy only a phone .
Alg. 2: i am curious to know if i could see here . **Alg. 3:** *am curious to know if* i am curious to know if i could see here .

Table 2: Consensus translations generated using BLEU risk over expected counts (Alg. 2), and using linear BLEU risk (Alg. 3). The use of linear BLEU risk in Algorithm 3 results in ill-formed consensus translations.

6. Summary

We have investigated different approaches to improve the MBRSC method described in [3]. First, we have proposed a new DP decoding algorithm to obtain the optimal consensus translation according to the original BLEU-based risk formulation. Then, we have studied a more efficient risk formulation based on the linear BLEU approximation proposed in [13]. Empirical results showed that the proposed DP decoding was able to obtain better-scoring higher-quality hypotheses than original gradient ascent search proposed in [3], and to do that with less temporal complexity. We have also shown that linear BLEU is not an adequate risk function for subsequence-combination methods due to the lack of n-gram count clippings in its formulation.

7. Acknowledgments

Work supported by the European Union Seventh Framework Program (FP7/2007-2013) under the CasMaCat project (grants agreement n^{o} 287576), by the Generalitat Valenciana under grant ALMPR (Prometeo/2009/014), and by the Spanish government under grant TIN2012-31723.

8. References

[1] O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2013 Workshop on Statistical Machine Translation," in *Proc. of the* 8^{th} *Workshop on SMT*, 2013, pp. 1–44.

[2] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. of the* 1^{st} *Workshop on Multiple Classifier Systems*, 2000, pp. 1–15.

[3] J. González-Rubio, A. Juan, and F. Casacuberta, "Minimum bayes-risk system combination," in *Proc. of the Association for Computational Linguistics*, 2011, pp. 1268–1277.

[4] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. of the Association for Computational Linguistics*, 2002, pp. 311–318.

[5] T. Nomoto, "Multi-engine machine translation with voted language model," in *Proc. of the Association for Computational Linguistics*, 2004, pp. 494–501.

[6] J. DeNero, S. Kumar, C. Chelba, and F. Och, "Model combination for machine translation," in *Proc. of the North American chapter of the Association for Computational Linguistics*, 2010, pp. 975–983.

[7] N. Duan, M. Li, D. Zhang, and M. Zhou, "Mixture model-based minimum bayes risk decoding using multiple machine translation systems," in *Proc. of the conference Computational Linguistics*, 2010, pp. 313–321.

[8] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 1997, pp. 347–354.

[9] S. Bangalore, "Computing consensus translation from multiple machine translation systems," in *Proc. of the IEEE workshop on Automatic Speech Recognition and Understanding*, 2001, pp. 351–354.

[10] A. Rosti, N. F. Ayan, B. Xiang, S. Matsoukas, R. Schwartz, and B. Dorr, "Combining outputs from multiple machine translation systems," in *Proc. of the North American Chapter of the Association for Computational Linguistics*, 2007, pp. 228–235.

[11] R. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1957.

[12] F. Jelinek, *Statistical methods for speech recognition*. Cambridge, MA, USA: MIT Press, 1997.

[13] R. Tromble, S. Kumar, F. Och, and W. Macherey, "Lattice minimum bayes-risk decoding for statistical machine translation," in *Proc. of the Empirical Methods in Natural Language Processing conference*, 2008, pp. 620–629.

[14] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE Transact. on Pattern Analisis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.

[15] R. Duda, P. Hart, and D. Stork, *Pattern classification*. Wiley, 2001.

[16] F. Och, "Minimum error rate training in statistical machine translation," in *Proc. of the Association for Computational Linguistics*, 2003, pp. 160–167.

[17] J. DeNero, D. Chiang, and K. Knight, "Fast consensus decoding over translation forests," in *Proc. of the Association for Computational Linguistics*, 2009, pp. 567–575.

[18] S. Kumar and W. Byrne, "Minimum bayes-risk decoding for statistical machine translation," in *Proc. of the North American Chapter of the Association for Computational Linguistics*, 2004, pp. 169–176.

[19] R. Stanley, *Enumerative combinatorics*. Cambridge University Press, 2002.

[20] X. He and K. Toutanova, "Joint optimization for machine translation system combination," in *Proc. of the Empirical Methods in Natural Language Processing conference*, 2009, pp. 1202–1211.

[21] C. Callison-Burch, P. Koehn, C. Monz, and J. Schroeder, "Findings of the 2009 Workshop on Statistical Machine Translation," in *Proc. of the* 4^{th} *Workshop on SMT*, 2009, pp. 1–28.

[22] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proc. of the Association for MT in the Americas*, 2006, pp. 223–231.

[23] Y. Zhang and S. Vogel, "Measuring confidence intervals for the machine translation evaluation metrics," in *Proc. of the* 10^{t} *conference on Theoretical and Methodological Issues in MT*, 2004.

Empirical Study of a Two-Step Approach to Estimate Translation Quality

Jesús González-Rubio[†], *J. Ramón Navarro-Cerdán*[‡], *Francisco Casacuberta*[†]

[†]D. de sistemas informáticos y computación [‡]Inst. Tecnológico de Informática
Universitat Politècnica de València, camino de Vera s/n, 46022 Valencia, Spain
jegonzalez@dsic.upv.es, jonacer@iti.upv.es, fcn@dsic.upv.es

Abstract

We present a method to estimate the quality of automatic translations when reference translations are not available. Quality estimation is addressed as a two-step regression problem where multiple features are combined to predict a quality score. Given a set of features, we aim at automatically extracting the variables that better explain translation quality, and use them to predict the quality score. The soundness of our approach is assessed by the encouraging results obtained in an exhaustive experimentation with several feature sets. Moreover, the studied approach is highly-scalable allowing us to employ hundreds of features to predict translation quality.

1. Introduction

Despite an intensive research in the last fifty years, machine translation (MT) systems are still far from perfect [1]. Hence, a desirable feature to improve their practical deployment is the capability of predicting at run-time[1] the reliability of the generated translations. This task, referred to as quality estimation [2] (QE), is becoming a crucial component in practical MT systems [3, 1]. For instance, to decide if an automatic translation is worth being supervised by a translator or it should be translated from scratch. Quality can be estimated at the word, sentence, or document level. Here, we focus on the estimation of sentence-level quality.

Sentence-level QE is typically addressed as a regression problem [4, 2]. Given a translation (and other sources of information), a set of features is extracted and used to build a model that predicts a quality score. This point of view provides a solid framework within which accurate predictors can be derived. However, several problems arise when applying this approach to predict the quality of natural language sentences. For example, while the concept of translation quality is quite intuitive, the definition of features that reliably account for it has proven to be elusive [4, 1]. Thus, in practice, feature sets contain a large number of noisy, collinear and ambiguous features that hinder the learning process of the regression models, e.g., due to the "curse of dimensionality" [5].

An interesting approach to overcome these problems is to conceive QE as a two-step problem. In a first step, a dimensionality reduction (DR) process strips out the noise present in the original features returning a reduced set of (potentially new) features. Then, the actual quality prediction is made from this reduced set. Typically, QE systems reduce the dimensionality by simply selecting a subset of the original features according to some relevance measure [2, 6, 7]. However, a recent study [8] have shown that DR methods based on a projection of the original features may be more effective. The intuition for this is clear, the new features extracted by a projection-based DR method summarize the "information" contained in the all the original features, in contrast, the information contained in the features discarded by a feature selection method is inevitably lost.

We work on the foundations of [8] and provide an exhaustive empirical study of the most successful QE approach described there. This approach (§2) involves a DR method based on a partial least squares [9] (PLS) projection of the data and a support vector machine [10] (SVM) as prediction model. We test this two-step QE approach in a wide variety of conditions (§3) where we compare the performance of PLS to the most widely-used projection-based DR approach, namely principal component analysis [11] (PCA). Empirical results (§4) show that PLS consistently outperformed PCA in prediction accuracy and feature reduction ratio. This latter result is particularly interesting because it allows us to apply QE in scenarios with strict temporal restrictions, for instance interactive machine translation tasks.

[1]That is, in the absence of reference translations.

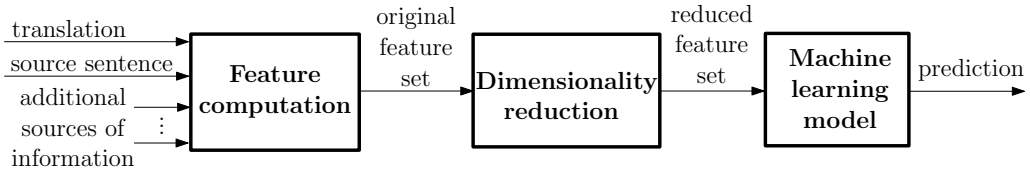


Figure 1: Dataflow of the studied two-step QE approach.

2. A Two-Step QE Approach

The method proposed in [8] divide QE ($\mathbb{R}^m \rightarrow \mathbb{R}$) into two sub-problems. First, the original m-dimensional set of features is projected into a new r-dimensional set of features ($\mathbb{R}^m \rightarrow \mathbb{R}^r, r < m$). Then, this reduced feature set is used to build a regression model that predicts the actual quality scores ($\mathbb{R}^r \rightarrow \mathbb{R}$). Figure 1 shows a diagram of this two-step training methodology. Next sections describe how to solve these two sub-problems.

2.1. Dimensionality Reduction

Typical approaches to reduce a set of noisy features involve the use of principal components analysis [11] (PCA). PCA projects the set of features into a set of principal components (PCs) where each PC explains the variability of the features in one principal direction. As a result, these PCs contain almost no redundancy but, since the PCA transformation ignore the quality scores to be predicted, they do not necessarily have to be the best features to perform the prediction.

Instead, we implement a feature reduction technique based on partial least squares [9] (PLS). PLS extracts a ordered set of latent variables (LVs) such that each of them accounts for the maximum possible covariability between the features and the scores to be predicted under the constraint of being uncorrelated with previous LVs. That is, LVs are uncorrelated as PCs do, and additionally, they explain as much as the variability in the quality scores as possible. As a result, usually few LVs than PCs are required to reach a certain accuracy.

Let $\{\mathbf{x}_i, y_i\}_{i=0}^n$ be a corpus with n samples where $\mathbf{x}_i$ are m-dimensional feature vectors, and y_i are quality scores. This corpus can be written in matrix form where symbol $^\intercal$ indicates the transpose of a matrix or vector:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^\intercal \\ \vdots \\ \mathbf{x}_n^\intercal \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{1}$$

Then, PLS constructs the following linear model where $\mathbf{b}$ is a vector of regressor coefficients, and $\mathbf{f}$ is a vector of zero-centered Gaussian errors:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{f} \tag{2}$$

PLS also defines two PCA-like transformations ($\mathbf{P}$ for $\mathbf{X}$, and $\mathbf{q}$ for $\mathbf{y}$) with $\mathbf{E}$ and $\mathbf{f}$ being the corresponding errors, and a linear relation $\mathbf{R}$ linking both blocks:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^\intercal + \mathbf{E} \quad \mathbf{y} = \mathbf{U}\mathbf{q}^\intercal + \mathbf{f} \qquad \mathbf{U} = \mathbf{T}\mathbf{R} \tag{3}$$

where matrices $\mathbf{T}$ and $\mathbf{U}$ are the projections of $\mathbf{X}$ and $\mathbf{y}$ respectively. The value of the regression coefficients $\mathbf{b}$ are finally computed as [9]:

$$\mathbf{b} = \mathbf{R}\mathbf{q}^\intercal \quad \text{where} \quad \mathbf{R} = \mathbf{W}(\mathbf{P}^\intercal\mathbf{W})^{-1} \tag{4}$$

where $\mathbf{W}$ is a weight matrix that accounts for the correlation between $\mathbf{X}$ and $\mathbf{U}$.

The columns in matrix $\mathbf{T}$ are the LVs of $\mathbf{X}$. Each of these LVs accounts for the maximum co-variability between $\mathbf{X}$ and $\mathbf{y}$ not explained by previous LVs. Therefore, similarly as it is usually done with PCA, we can collect the first r LVs and use them to represent the original m-dimensional feature set. Given that $r < m$, and that the LVs are orthogonal by definition, we are simultaneously addressing the "curse of dimensionality" and reducing the noise present in the original features. Moreover, the reduced set also explains most of the variability in the quality scores to be predicted.

In the experiments, we used the `pls` library [12] of the R toolkit. The dimension of the reduced set r is one of the meta-parameters of the studied QE approach.

PLS can be directly used as a predictor model (see Equation (2)). However, its simple linear model is not adequate to model the nonlinear relation that may exist between the features and the quality scores. Preliminary experiments confirmed this intuition.

2.2. Prediction Model

Once the reduced feature set is extracted, a support vector machine (SVM) is used predict the quality scores ($\mathbb{R}^r \rightarrow \mathbb{R}$). We choose SVMs because they have shown good prediction accuracy and robustness when dealing with noisy data in a number of tasks.

SVMs, first proposed for classification problems by Cortes and Vapnik [10], are a class of machine learning models that are able to model nonlinear relations between the features and the values to be predicted. Prior to any calculation, SVMs project the data into an alternative space. This projection, defined by a kernel function, may be nonlinear; thus, though a linear relationship is learned in the projected feature space, this relationship may be nonlinear in the original space. Following previous works on QE [2], we use SVMs with a radial basis kernel as implemented in the `LibSVM` package [13]. Values γ, ϵ, and C are additional meta-parameters to be optimized.

3. Experimental Setup

3.1. Corpus

We computed quality scores for the English-Spanish news evaluation data used in the QE task of the 2012 workshop on statistical MT [1] (WMT12). The Spanish translations were generated by a phrase-based MT system trained on the Europarl and News Commentaries corpora as provided for the WMT12 translation task. Evaluation data contains 1832 translations for training, and 422 translations for test. The quality score of each translation $\{y \in \mathbb{R} \mid 1 \leq y \leq 5\}$ is computed as the average of the scores given manually by three different experts in terms of post-editing effort:

5: The translation requires little editing to be publishable
4: 10%–25% of the translation needs to be edited
3: 25%–50% of the translation needs to be edited
2: 50%–70% of the translation needs to be edited
1: The translation must be translated from scratch

3.2. Feature Sets

We conducted QE experiments with several feature sets submitted to the WMT12 QE task[2]. These sets allow us to test our approach under a wide variety of conditions. Table 1 displays, for each set, the number of features, whether or not the features are result of a feature selection process, the percentage of features in the training partition that are collinear with the rest of features (redundancy), and the percentage of features in the training partition that are constant, and hence, irrelevant to perform the prediction. We estimated the degree of collinearity of each feature by its condition number considering a value above 100 to denote collinearity [14]

[2]These are available in https://github.com/lspecia/QualityEstimation.

Name	#features	feature selection?	collinear features	constant features
DCU-SYMC	308	no	34.6%	0.7%
LORIA	49	yes	12.2%	0.0%
SDLLW	15	yes	0.0%	0.0%
TCD	43	no	18.6%	0.0%
UEDIN	56	no	5.5%	1.8%
UPV	497	no	54.3%	6.8%
UU	82	no	7.5%	2.5%
WLV-SHEF	147	no	21.0%	2.7%

Table 1: Main properties of the feature sets. We estimated the collinearity with the condition number [14].

We consider the feature sets as independent corpora provided by an external agent. Hence, and due to space limitations, we only provide a brief description of each set; an exhaustive description can be found in the corresponding citation. Many of the sets include the 17 baseline features provided by the organizers [1].

DCU-SYMC: [15] 308 features including features based on latent Dirichlet allocation; source grammatical features from the TreeTagger part-of-speech tagger, an English grammar, the XLE parser, and the Brown re-ranking parser; and target TreeTagger features.

LORIA: [6] 66 features including the baseline features, and features based on cross-lingual triggers.

SDLLW: [7] 15 features exhaustively selected from an original set of 45 features: the 17 baseline features, 8 features based on decoder information, and 20 features based on n-gram precisions and word alignments.

TCD: [16] 43 features including the baseline features, and features based on similarity measures with respect to the Google n-grams data set.

UEDIN: [17] 56 features including the baseline features and features based on named entities, morphological information, lexicon probabilities, word-alignments, and sentence and n-grams similarities.

UPV: [18] 497 features including the baseline features and features based on word-level quality scores.

UU: [19] 82 features computed from syntactic, constituency, and dependency trees.

WLV-SHEF: [20] 147 features based on part-of-speech information, subject-verb agreement, phrase constituency and target lexicon analysis.

3.3. Experimental Methodology

For each feature set, a QE system was built following the two-step methodology described in §2 and depicted

Feature set	Baseline		PCA			Our approach		
	RMSE	#features	RMSE	#features		RMSE	#features	
DCU-SYMC	0.71±0.02	308	0.70±0.02	82	(26.6%)	**0.62±0.02***	**28**	**(9.1%)**
LORIA	**0.72±0.03**	49	0.75±0.01	43	(87.7%)	**0.72±0.02**	**10**	**(20.4%)**
SDLLW	**0.67±0.02**	15	**0.67±0.02**	15	(100.0%)	**0.67±0.02**	**10**	**(66.7%)**
TCD	0.76±0.01	43	0.74±0.02	24	(55.8%)	**0.72±0.02**	**15**	**(38.9%)**
UEDIN	0.72±0.03	56	0.71±0.02	43	(76.8%)	**0.69±0.02**	**8**	**(14.3%)**
UPV	0.74±0.02	497	0.69±0.02	99	(19.9%)	**0.62±0.02***	**58**	**(11.7%)**
UU	0.72±0.02	82	0.68±0.02	74	(90.2%)	**0.67±0.02**	**29**	**(35.4%)**
WLV-SHEF	0.71±0.02	147	0.71±0.02	91	(61.9%)	**0.65±0.02***	**25**	**(17.0%)**

Table 2: RMSE and number of LVs obtained by cross-validation for the different feature sets. In parenthesis, we show the number of LVs as a percentage of the original features. Baseline denotes a system trained with the whole feature set. PCA denotes a system built using PCA instead of PLS. Best mean RMSE values and lowest number of features are displayed boldface. Asterisks denote a statistically better result than *both* the other two systems (95% confidence).

in Figure 1. All features were standardized by subtracting the feature mean from the raw values, and dividing the difference by the corresponding standard deviation.

The number of LVs (r) was optimized by ten-fold cross-validation using the training partitions (1832 samples). Each cross-validation experiment took eight folds for training (dev-train), one held-out fold for development and the other held-out fold for test (dev-test). We used the dev-train folds to estimate a PLS model. Then, this model was used to extract the r LVs of dev-train, and of the separated development fold and the dev-test fold. Next, we used the reduced dev-train folds to estimate an SVM model, the reduced development fold to optimize the SVM meta-parameters (γ, ϵ, and C), and the reduced dev-test fold to test the optimized SVM model. The result of each complete cross-validation experiment was the averaged prediction accuracy on the ten held-out dev-test folds. The number of LVs was selected to optimize this average accuracy.

Once the number of LVs was fixed, we built a new prediction model with the whole training partition optimizing the SVM meta-parameters by cross-validation. Finally, we used this optimized SVM model to predict the quality scores of the test partitions (422 samples).

3.4. Assessment Criteria

We measure the accuracy of a QE system by the deviation of its predictions $\hat{\mathbf{y}} = \{\hat{y}_1, \ldots, \hat{y}_n\}$ respect to the reference quality scores $\mathbf{y} = \{y_1, \ldots, y_n\}$. Following previous QE works [2, 1], we calculate the root-mean-squared error (RMSE) between them:

$$\text{RMSE}(\hat{\mathbf{y}}, \mathbf{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (5)$$

where n is the number of samples. RMSE quantifies the average error of the estimation with respect to the actual quality score. I.e. the lower the value, the better the performance of the QE system.

Additionally, we perform different significance tests for the reported RMSE results. On the one hand, we obtain confidence intervals for the averaged cross-validation test results with Student's t-tests [21]. On the other hand, we use paired bootstrap re-sampling [22] to measure the significance of the RMSE differences observed between the different methods in the test sets.

4. Results

We now present the results of the empirical evaluation of the studied QE approach. First, we predicted quality scores for each of the feature sets described in §3.2. Then, we took advantage of the scalability of the studied QE approach using jointly all the features in those sets to perform the prediction.

4.1. Results for the Individual Feature Sets

Table 2 shows the cross-validation results (RMSE and number of LVs) obtained for the different feature sets. As a comparison, we present results for SVMs trained with all the features in each set (Baseline), and for systems built using the widespread PCA instead of PLS in the studied two-step training methodology.

We can observe that the studied approach consistently obtained equal or better prediction accuracy (RMSE) than the baseline systems. Additionally, the number of LVs used to build the final SVMs was much lower than the number of original features. The size of the reduced sets varied between two thirds and one tenth

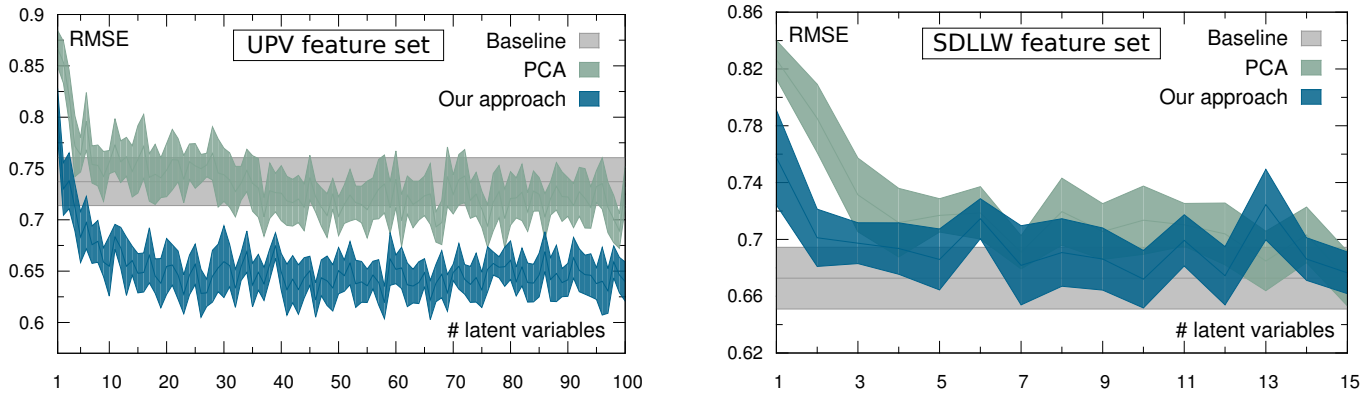


Figure 2: Cross-validation learning curves (RMSE and 95% confidence interval) for two representative feature sets: the highly-redundant UPV set (left), and the concise SDLLW set (right). Baseline denotes the RMSE of systems trained with the whole original feature sets: 497 features for UPV set, and 15 features for SDLLW set.

of the original features. This reductions are roughly related with the percentage of collinear and constant features in Table 1. In comparison to PCA, the studied DR technique, PLS, was able to obtain better prediction accuracy with less features. Usually, the number of LVs is less than half the number of PCs.

These result indicate that the studied QE approach was indeed able to strip out the noise present in the original features. Additionally, the DR technique based on PLS projections showed a better performance (both in prediction accuracy and reduction ratio) that the commonly-used PCA. As a result, even for highly-engineered features sets such as SDLLW [7] that contain no collinear or redundant features, our approach was able to obtain a more compact feature set (10 LVs) that still retained the prediction potential of the whole original set (15 features).

Next, to better understand the influence of the number of LVs in the results, Figure 2 displays the prediction accuracy as a function of the number of features for two prototypical feature sets: the highly noisy and collinear UPV set, and the low redundant SDLLW set.

The prediction accuracy of our method for the UPV feature set (left panel in Figure 2) rapidly improved as more LVs were considered. With only 5 LVs, prediction accuracy already statistically outperformed the baseline (497 features), and it reached its top performance for 58 LVs. As we considered more LVs (for simplicity the graph only shows up to 100 LVs), prediction error steadily increased which was indicative of over-training. Thus, we chose 58 as the optimum number of variables for the UPV set. The quite large RMSE reduction respect to the baseline can be explained by the ability of our approach to strip out the great amount of noise present in the original UPV set, see Table 1. Regarding PCA, it was consistently outperformed by our approach and only slightly improved the RMSE score of the baseline system.

For the concise SDLLW feature set (right panel in Figure 2), our system showed approximately the same behavior: prediction accuracy rapidly improved up to a point from where the performance remains approximately stable. In this case, 10 was the optimal number of LVs. In contrast to the UPV set, our approach could not improve Baseline performance which is reasonable since SDLLW is a very clean set with no redundant or irrelevant features (see Table 1) that could hinder the learning process. Nevertheless, our method was able to obtain the same prediction accuracy as Baseline with only two thirds of the original features.

In a following experiment, we built QE systems with the whole training partitions and the optimal number of LVs estimated in the previous cross-validation experiments. The SVM meta-parameters (γ, ϵ, and C) were optimized by standard cross-validation and the optimized models were used to predict the quality scores of the test partitions. Note that due to variations in the learning procedures, Baseline results may differ from those reported in the WMT12 QE task [1] .

Table 3 displays, for each feature set, the RMSE obtained by our approach in the test partition. We also show baseline results for SVMs built with all the features in each set, and for systems that used PCA instead of PLS to reduce the dimensionality. RMSE confidence intervals for Baseline, PCA and our approach always overlapped but the observed differences were still statistically significant for a number of sets: for DCU-SYMC, Baseline obtained a statistically bet-

Feature set	Baseline	PCA	Our approach
DCU-SYMC	**0.87±0.07***	1.01±0.07	0.96±0.08
LORIA	**0.84±0.06**	0.87±0.06	0.85±0.06
SDLLW	**0.76±0.05**	0.77±0.05	**0.76±0.05**
TCD	**0.82±0.06**	1.00±0.05	0.83±0.06
UEDIN	0.86±0.06	**0.85±0.05**	0.86±0.05
UPV	0.82±0.06	0.83±0.05	**0.78±0.05***
UU	**0.81±0.05**	**0.81±0.05**	0.82±0.06
WLV-SHEF	0.84±0.05	0.84±0.05	**0.82±0.05***

Table 3: RMSE and 95% confidence intervals of the predictions for the test partitions. Best mean results are displayed boldface. Asterisks denote a significant difference in performance (paired re-sampling, 95% confidence) respect to *both* the other two methods.

ter result than PCA and our approach; for LORIA and TCD, no statistically significant difference was observed between our approach and Baseline but both systems obtained a statistically better result than PCA; for UPV and WLV-SHEF, our approach statistically outperformed the other two methods; and for SDLLW, UEDIN and UU, no significant differences were found.

These were quite surprising results. Given the encouraging RMSE improvements observed in cross-validation (see Table 2), we expected to obtain similar differences over Baseline in test. We followed a careful cross-validation training process (see Section 3.3) where each experiment was evaluated in a held-out test fold used neither to reduce the dimensionality nor to estimate the prediction model. Therefore, we hypothesized that the explanation for the results in Table 3 was that the training partitions were not representative of the test partitions. We evaluated this hypothesis by means of a series of multivariate Hotelling's two-sample T^2 tests [23]. The objective of these tests is to determine if two samples (in our case the values of the features in the training and test partitions) have been sampled from the same population or not. The results of the tests indicated that, for all feature sets, the training and test partitions were indeed statistically different ($p < 0.01$). In contrast, no statistical difference was found, for any of the feature sets, between the dev-train and dev-test folds used in the cross-validation training process.

In a more fine-grained analysis, we study individually the features in each set. The results of a series of Student's two-sample t-tests [21] indicated that most of the features did exhibit statistically different values ($p < 0.01$) between training and test. E.g., the value

DCU-SYMC	45.1%	UEDIN	48.1%
LORIA	24.5%	UPV	67.4%
SDLLW	73.3%	UU	38.8%
TCD	30.2%	WLV-SHEF	28.6%

Table 4: Ratio of the features in each set that have significantly different values in the training and test partitions. These ratios reduce to about 1% in the dev-train and dev-test cross-validation folds. Significance computed by Student's two-sample t-test (99% confidence).

one of these "mismatched" features in the UPV set was $\mu = 1.7$ ($\sigma = 1.4$) in training, and $\mu = 0.9$ ($\sigma = 0.8$) in test. In contrast, only about 1% of the features exhibit different values between the cross-validation dev-train and dev-test folds. Table 4 displays, for each set, the percentage of "mismatched" features between the training and test partitions.

This mismatch can be partially explained by the fact that the training and test partitions contain news texts of different years [1], but we still consider that the main issue is the size (only 1832 samples) of training partitions that did not adequately represent test partitions. However, both our approach and the baseline systems had to deal with this mismatch, so, why our method and PCA seemed to be more heavily penalized than Baseline?

The projection of the features is computed based on the training data. Thus, if the training partition is not representative of the test partition, the reduced feature sets will be projected in a "direction" that may penalize the prediction accuracy for the test set. That is, crucial information to predict the quality scores of the test partition may be stripped out. This drawback is common to any dimensionality reduction technique as exemplified by the also poor test results (Table 3) obtained by PCA.

The conclusion that can be extracted from these results is that the use of feature reduction implies a greater risk of over-training the prediction system. This effect particularly important if training data is scarce but it is mitigated as more training data is available. Thus, given the encouraging cross-validation results in Table 2, better prediction accuracy could be expected in test whenever an adequate training partition is provided.

Under the assumption that the original features can be computed in advance, a complimentary advantage of the studied two-step QE approach is that it allows us to build more time-efficient QE systems. Figure 3 displays the time required to build an SVM model (including meta-parameter optimization) and obtain the test predictions as a function of the number of features

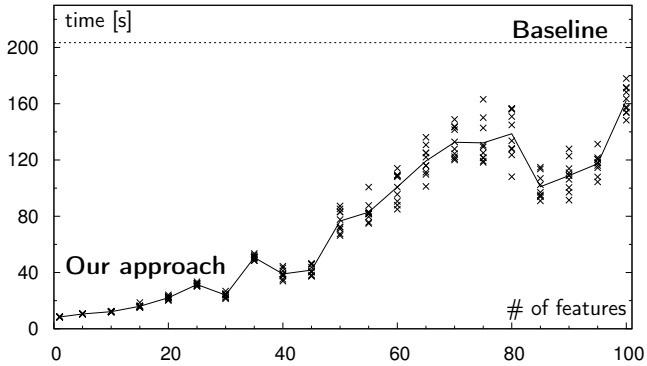


Figure 3: Operating time (training plus prediction) of the SVM model as a function of the number of features used to built the model. Baseline system was trained with the 147 original features of the WLV-SHEF set.

used to train the model. Specifically, we built QE systems with an increasing number of LVs extracted from the WLV-SHEF feature set. Each point in the figure is the average time of ten experiments. Results show how operating times increased with the number of LVs. For instance, the operating time of the baseline model trained with the original 147 features (0.84 RMSE) was $\sim$200 seconds, while the operating time of the system built with the 14 LVs extracted by PLS (0.82 RMSE) was only $\sim$15 seconds which represents one order of magnitude less operating time. Hence, our approach is well-suited to be applied to scenarios, such as interactive MT [3], with strict temporal restrictions.

4.2. Exploiting the scalability of our approach

Results in the previous section have shown that the studied QE approach was able to extract the relevant prediction information from different sets of noisy features. We now take a further step in this direction and present results where all the features used in the previous experiments are joined together to create an extremely high-dimensional feature set from which to predict quality scores. This aggregated set, denoted by ALL, contains 1197 features for each translation; approximately 55% of them being collinear with the rest.

Figure 4 shows cross-validation prediction accuracy (RMSE and 95% confidence interval) of the studied QE approach as a function of the number of LVs. Again, we also display results for a baseline SVM model built using all the features, and for a system built using PCA instead of PLS. Our approach obtained a score of 0.45 ± 0.01 RMSE with only 86 LVs. This result represents a 30% reduction relative to the baseline RMSE calculated with 1197 features. Regarding PCA, it barely

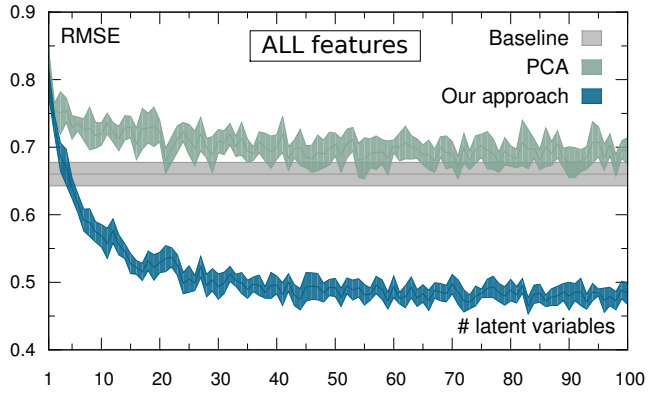


Figure 4: Cross-validation learning curve for the high-dimensional (1197 features) ALL set.

reached Baseline performance. These results indicate that our approach was able to exploit the information contained in the ALL set to improve prediction accuracy. In contrast, both Baseline and PCA were unable to adequately manage the huge number of noisy and collinear features. Additionally, the operating time of the Baseline systems was $\sim$ 23 minutes, while it reduced to $\sim$2 minutes when we used the optimal 86 LVs.

Test results were again quite disappointing: 1.4 ± 0.1 RMSE of our approach versus 0.78 ± 0.06 RMSE of Baseline and 0.81 ± 0.07 of PCA. We hypothesize that the clearly worse result of our approach in this case was due to the larger number features. As more features are available, our system can generate more "specialized" LVs. Given that the training data does not adequately represents the test data (see discussion in §4.1), this better projection (as shown in Figure 4) actually hinders prediction accuracy in the test set.

5. Summary

We have described an empirical study of a two-step QE approach specifically designed to manage the noisy features usually derived from natural language sentences. This approach, first described in [8] implements a method based on PLS to extract, from the original features, the LVs that actually govern translation quality, and an SVM model to actually predict the quality scores from these LVs.

Empirical cross-validation results showed that the studied QE approach was able to obtain very large feature reduction ratios, and at the same time, it usually outperformed systems built with all the original features and systems that use PCA instead of PLS to reduce the dimensionality. Unfortunately, results in the held-out test partitions were disappointing. The results of differ-

ent statistical tests seem to indicate that this was due to the small size of the training partitions. Hence, larger RMSE improvements could be expected in test whenever a representative training partition is provided.

A complimentary advantage of the studied QE approach is its time-efficiency. This fact makes our approach well-suited to be deployed in scenarios with strict temporal restrictions, such as interactive MT systems. Alternatively, we could take advantage of this efficiency to predict translation quality from huge sets of features. Results in this direction show that our approach was able to efficiently manage more than a thousand features largely improving prediction accuracy.

6. Acknowledgments

Work supported by the European Union 7^{th} Framework Program (FP7/2007-2013) under the CasMaCat project (grans agreement n$^{\text{o}}$ 287576), by Spanish MICINN under grant TIN2012-31723, and by the Generalitat Valenciana under grant ALMPR (Prometeo/2009/014).

7. References

[1] C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2012 workshop on statistical machine translation," in *Proc. of the* 7^{th} *Workshop on SMT*, 2012, pp. 10–51.

[2] L. Specia, M. Turchi, N. Cancedda, M. Dymetman, and N. Cristianini, "Estimating the sentence-level quality of machine translation systems," in *Proc. of the European Association for Machine Translation*, 2009, pp. 28–35.

[3] J. González-Rubio, D. Ortiz-Martínez, and F. Casacuberta, "Balancing user effort and translation error in interactive machine translation via confidence measures," in *Proc. of the Association for Computational Linguistics*, 2010, pp. 173–177.

[4] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing, "Confidence estimation for machine translation," in *Proc. of the conference on Computational Linguistics*, 2004, pp. 315–321.

[5] R. Bellman, *Adaptive control processes: a guided tour*. Princeton University Press, 1961.

[6] D. Langlois, S. Raybaud, and K. Smaïli, "LORIA system for the WMT12 quality estimation shared task," in *Proceedings of the* 7^{th} *Workshop on SMT*, 2012, pp. 114–119.

[7] R. Soricut, N. Bach, and Z. Wang, "The SDL Language Weaver systems in the WMT12 quality estimation shared task," in *Proceedings of the* 7^{th} *Workshop on SMT*, 2012, pp. 145–151.

[8] J. González-Rubio, J. R. Navarro-Cerdán, and F. Casacuberta, "Dimensionality reduction methods for machine translation quality estimation," *Machine Translation*, pp. 1–21, 2013.

[9] H. Wold, *Estimation of Principal Components and Related Models by Iterative Least squares*. Academic Press, 1966, pp. 391–420.

[10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[11] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.

[12] B. Mevik, R. Wehrens, and K. H. Liland, *PLS: Partial Least Squares and Principal Component regression*, 2011, R package version 2.3-0.

[13] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

[14] E. W. Cheney and D. R. Kincaid, *Numerical Mathematics and Computing*. Brooks/Cole, 2012.

[15] R. Rubino, J. Foster, J. Wagner, J. Roturier, R. Samad Zadeh Kaljahi, and F. Hollowood, "Dcu-symantec submission for the WMT 2012 quality estimation task," in *Proc. of the* 7^{th} *Workshop on SMT*, 2012, pp. 138–144.

[16] E. Moreau and C. Vogel, "Quality estimation: an experimental study using unsupervised similarity measures," in *Proceedings of the* 7^{th} *Workshop on SMT*, 2012, pp. 120–126.

[17] C. Buck, "Black box features for the WMT 2012 quality estimation shared task," in *Proceedings of the* 7^{th} *Workshop on SMT*, 2012, pp. 91–95.

[18] J. González-Rubio, A. Sanchís, and F. Casacuberta, "PRHLT submission to the WMT12 quality estimation task," in *Proceedings of the* 7^{th} *Workshop on SMT*, 2012, pp. 104–108.

[19] C. Hardmeier, J. Nivre, and J. Tiedemann, "Tree kernels for machine translation quality estimation," in *Proceedings of the* 7^{th} *Workshop on SMT*, 2012, pp. 109–113.

[20] M. Felice and L. Specia, "Linguistic features for quality estimation," in *Proceedings of the* 7^{th} *Workshop on SMT*, 2012, pp. 96–103.

[21] W. Gosset, "The probable error of a mean," *Biometrika*, no. 1, pp. 1–25, 1908.

[22] Y. Zhang and S. Vogel, "Measuring confidence intervals for the machine translation evaluation metrics," in *Proc. of the Conference on Theoretical and Methodological Issues in Machine Translation*, 2004.

[23] T. Anderson, *An Introduction to Multivariate statistical Analysis*. New York: Wiley, 1958.

The 2013 KIT Quaero Speech-to-Text System for French

Joshua Winebarger, Bao Nguyen, Jonas Gehring, Sebastian Stüker, and Alexander Waibel

Institute for Anthropomatics
Karlrsuhe Institute of Technology, Karlsruhe
{joshua.winebarger|quoc.nguyen|jonas.gehring|sebastian.stueker|alex.waibel}@kit.edu

Abstract

This paper describes our *Speech-to-Text* (STT) system for French, which was developed as part of our efforts in the Quaero program for the 2013 evaluation. Our STT system consists of six subsystems which were created by combining multiple complementary sources of pronunciation modeling including graphemes with various feature front-ends based on deep neural networks and tonal features. Both speaker-independent and speaker adaptively trained versions of the systems were built. The resulting systems were then combined via confusion network combination and cross-adaptation. Through progressive advances and system combination we reach a word error rate (WER) of 16.5% on the 2012 Quaero evaluation data.

1. Introduction

1.1. The Quaero Speech-to-Text Task

Quaero (http://www.quaero.org) is a French research and development program with German participation. The focus is to develop multimedia and multilingual tools with professional and general public applications in such domains as automatic extraction, analysis, classification, and exploitation of information. The vision of Quaero is to provide public and professional users with the means to access various information types and sources in digital form. Quaero proposes to achieve this by creating a framework for collaboration between complementary technological ventures such as businesses, public research institutions, and universities through competitive evaluations and sharing of the research thereby created in a process called "coopetition." Partners also collaborate on advanced demonstrations and prototypes and work to develop and commercialise the resulting applications and services.

One of the technologies researched within Quaero is *Automatic Speech Recognition*, i.e. the automatic transcription of human speech into written form. This is known as the speech-to-text task. In line with the concept of coopetition, evaluation of ASR technological development in Quaero is done once a year. The domain is a mix of broadcast news and broadcast conversational speech, the latter of which is more challenging for automatic recognisers than read speech due to the presence of disfluencies and non-speech events such as music and spontaneous human noises. The number of languages included in the program has increased, as has the expected state-of-the-art recognition system performance. Being a French project with European orientation, French is naturally among the languages evaluated. The fall 2013 evaluation was the fifth and final full-scale evaluation of ASR within Quaero. The test data for the evaluation consisted of audio from various web sources including broadcast news, video blogs, and lectures. At the time of this writing the evaluation was not yet completed. Therefore this paper reports our results on the 2012 evaluation data, which we used as our development set.

1.2. Paper Structure

The paper is structured as follows. Section 2 describes the acoustic data and training techniques of our system. We describe the front-end processing used, including deep neural networks and tonal features. Our efforts to develop diversified pronunciation modeling are the focus of Section 3. We give special attention to the use of graphemes. Then, we make a detailed description of our language model and its development in Section 4. In Section 5 we describe our overall recognition setup used in the evaluation and give performance figures for the 2012 evaluation data. Section 6 describes experiments done in the development of our system. Finally we discuss opportunities for future work and conclude the paper in Section 7.

2. Acoustic Modeling

We trained several acoustic models based on different pronunciation dictionaries and feature front-ends. The pronunciation modeling aspect is described in detail in Section 3. Each pronunciation model (dictionary) was essentially based on its own phoneme set. The feature front-ends can be generally described as deep bottlneck features based on either *i)* 40 log mel filter bank coefficients (lMEL) or *ii)* a stacked combination of 20 mel-frequency cepstal coefficients (MFCC,) 20 warped minimum variance distortionless response coefficients (MVDR,) and tonal features (a setup we call *M2+T*), described in Section 2.3.

All acoustic models are based on HMMs, whose states correspond to generalized quinphones with three states per phoneme, and a left-to-right topology without skip states. The generalized quinphones were found by clustering the quinphones in the training data using a decision-tree. We found 8,000 acoustic models performed best in all subsystems except the grapheme subsystem, for which 12,000 models performed best. The models were trained using incremental splitting of Gaussians (also known as merge and split or MAS training.) For all models we then estimated one global semi-tied covariance (STC) matrix after LDA [1], and refined the models with two iterations of viterbi training. All models use vocal tract length normalization (VTLN.) For a second-pass decoding (see Section 5) speaker adaptive models are trained using feature space constrained MLLR [2, 3]. For certain systems we extended the expected maximisation of the models with discriminative training based on the boosted Maximum Mutual Information Estimation (bMMIE) criterion [4], which saw reductions in word error rate (WER) of between 2-2.5% (relative) compared to the maximum likelihood systems.

2.1. Training Data

Each subsystem was trained on 268 hours of speech coming from the Broadcast News (BN) and Broadcast Conversation (BC) domain. We used the Quaero training data from 2009-2011 as well as data from the Ester campaign [5]. Both datasets provide manual transcripts and speaker clustering. The Quaero data can be divided into portions for which the transcripts are "fast" or carefully annotated. Those using careful annotations have speakers identified by name, even across shows and audio files, whereas "fast" annotated transcripts use automatic speaker annoatations. Also, the carefully annotated transcripts have a more comprehensive and detailed transcription of noises, disfluencies, and hesitations. We used a technique for filtering this acoustic data by decoding on it, which is described in Section 6.1. Before filtering we had 194.1 hours of Quaero data and 107.8 hours of Ester data. After filtering we had 187.7 hours of Quaero data and 80.3 hours of Ester data, which was used for training.

2.2. Deep Neural Networks as Features

Bottleneck features (BNFs) from multilayer perceptrons have become a staple component in ASR, due to their discriminative power and robustness to speaker and environment variations. Gehring et. al. recently introduced a deep bottlneck feature (DBNF) archictecture based on deep neural networks (DNNs) consisting of many hidden layers, which was shown to achieve significant reductions in WER. [6, 7]

For our French system we trained deep bottleneck feature networks for each subsystem from the best existing MFCC system alignment. Input to the network takes place on an input layer accepting stacked MFCC, MVDR and Tonal features or log mel coefficients. The network consists of five

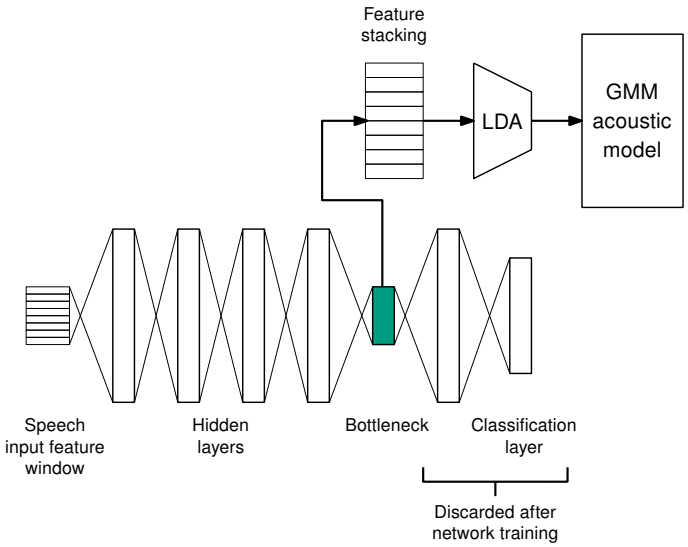


Figure 1: *Deep bottleneck architecture used for feature extractation in our systems*

fully-connected hidden layers containing 1,200 units each, followed by the bottleneck layer with 42 units, a further hidden layer and the final layer, as can be seen in Figure 1.

Layers prior to the bottleneck are pre-trained in a layer-wise, unsupervised manner as a stack of denoising auto-encoders [8]. After the stack of auto-encoders has been pre-trained, the bottleneck layer, the next hidden layer, and the classification layer are intialized with random weights and connected to the hidden representation of the top-most auto-encoder. The network is then trained with supervision to estimate the context-dependent HMM polyphone states. Fine tuning is performed for 14-18 epochs using the "newbob" learning rate schedule which starts with a high learning rate until the increase in accuracy on a validation set drops below a set threshold. The learning rate is then halved for each epoch until improvement in validation accuracy drops below a second threshold, at which point learning is stopped. The activations of the 42 bottleneck units are stacked over an 11- to 13-frame context window and reduced to a dimensionality of 42 using LDA.

Our context-dependent systems were then trained with these networks using the existing polyphone tree and alignment computed without DBNFs. Relative to MFCC, we saw an average word error rate reduction of approximately 22% for systems using lMEL DBNFs and 24% for those using DBNFs based on MFCC and MVDR. This is comparable to gains we saw in development of recognisers for other languages, where the same or similar DBNF architecture was employed.

2.3. Tonal Features

Conventional wisdom in ASR asserts that pitch or "tonal" information is not helpful in building speech recognisers for

non-tonal languages (such as French.) However it was recently shown that pitch information can be integrated into an ASR in a manner that improves recognition accuracy for both tonal and non-tonal languages [9]. Fundamentally different from spectral features, which capture the envelope of the speech signal, pitch features capture variations in the fundamental frequency of the speaker's voice. Our DBNFs based on a concatenation of MFCC and MVDR coefficients incorporated two such tonal features derived from the pitch of the speech signal. These are the pitch and Fundamental Frequency Variation (FFV.)

We extract pitch features according to the method in [10]. First, a cepstrogram is computed with a window length of 32 ms. We detect the position of the maximum of all cepstral coefficients starting with the 30^{th} coefficient. Dynamic programming is then used to find a path that maximises the correlation between coefficients subject to constraints such as the maximum pitch change per unit time. Additionally we consider the position of the three left and right neighbours, as well as their first and second derivatives, resulting in seven pitch coefficients.

FFV features [11], typically used for tasks such as speaker verification, have the advantage that no explicit segmentation into speech and silence (for which pitch is undefined) is necessary. The change in fundamental frequency is not computed by tracking a single value of F_0 over time. Rather a "vanishing point product" is computed between two feature vectors obtained from two asymmetric windows covering the left-half and right-half portion of the general feature window. This vanishing point is equivalent to an inner product between left and right spectrums F_L and F_R, where F_L or F_R are dilated with respect to one another by positive or negative values of τ, respectively. This vanishing point product is depicted graphically in Figure 2. Afterwards, a filterbank is applied which attempts to capture meaningful prosodic information. The filter bank contains a trapezoidal filter for perceptually "flat" pitch, two trapezoidal filters for "slowly changing" (rising and falling) pitch, and two additional trapezoidal filters for "rapidly changing" pitch. In addition, the filterbank contains two rectangular extremity filters, as unvoiced frames have flat rather than decaying tails. This filterbank reduces the input space to 7 scalars per frame, which we use as additional "FFV" features in the final input vector. Previous experiments showed that the best way to integrate these features is through their concatenation with the MFCC and MVDR coefficients in the input vector for DBNF training. [9]

By concatenating tonal features in a 32 millisecond window with MFCC and MVDR coefficents (*M2+T*) for the input layer of our DBNFs, we reduced our error rate on the 2011 Quaero evaluation set by an additional 3% relative to MFCC and MVDR (*M2*) alone. This is comparable to the 3% relative improvement seen for KIT's English system developed for IWSLT. This 3% for non-tonal languages can be compared with the 5% relative improvement seen for Vietnamese, a tonal language.

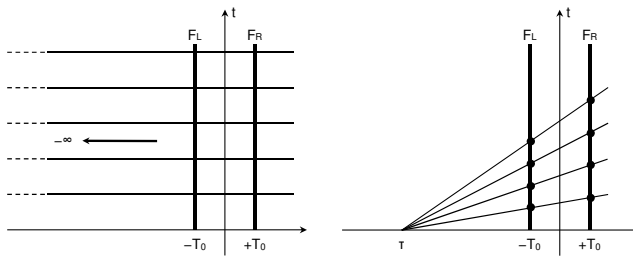


Figure 2: *Visualization of the vanishing point product employed in FFV. When $\tau \to \pm\infty$, the vanishing point product reduces to the standard inner product, shown on the left. On the right, we see F_R dilated by a negive value of τ.*

3. Pronunciation Modeling

In an effort to develop subsystems that produce diverse, complementary output for system combination and cross adaptation we employed different pronunciations modelings. For training and testing, we used pronunciation dictionaries based on four sources:

1. The popular BDLex lexicon, which gives a wide range of pronunciation variants
2. The Globalphone dictionary
3. A rule-based pronunciation generation called text2phone, typically used for TTS applications [12]
4. A pseudo-grapheme-based approach

After a straightforward mapping, the BDLex and Globalphone dictionaries shared essentially the same phone set. That of BDLex contains 45 phones, among them five noise phonemes: hesitation, incomplete words, human noises such as coughing, non-human noises such as music, and a catch-all noise. The Globalphone phone set adds a voiceless glottal fricative *h*, an additional open-middle vowel, and a breath noise. It also differs slightly in the classification of phones. For the text2phone system we used a different, 41-element set of phones, among them the same noise phones as that of BDLex.

For the first two sources of pronunciation, missing pronunciations were generated automatically using grapheme-to-phone models as described in [13]. Acoustic models for systems using the first two pronunciation sources were initialized by bootstrapping from German models using a manually created mapping.

While subsystems based on the BDLex dictionary generally yielded the best performance, we found that the system combination benefited from the inclusion of the output of each additional subsystem in the combination.

3.1. Pseudographeme System

In a traditional grapheme-based system, the symbols of the written word are used as the sub-units of pronunciation rather than phonemes. The feasibility of using graphemes instead of phonemes in ASR has been shown in several different works [14, 15, 16]. It was also shown that the combination of a grapheme system with phoneme systems lead to a significant reduction in word-error rate [17].

While French orthography is relatively regular vis à vis a language like English, the mapping between sounds and graphemes is not bijective, which is to say that the correspondence between graphemes and phonemes can be weak. Often, clusters of graphemes produce the same sounds as other, shorter ones, such as "-ai" and "-é", which both correspond to the IPA [e]. We handle this weakness by using single or multiple graphemes as the base units of pronunciation. Our set of grapheme-phones contained 49 elements, among them the same five noise phones as in the BDLex system. Using knowledge of French pronunciation we wrote simple mappings determining whether a certain sequence of graphemes should map to one unit or stand as separate units of pronunciation. These mappings consist of a list of grapheme sequences to be merged. We scan from left to right in a word and seek the longest matches possible in the list. The rules are kept simple in that we do not attempt to merge the graphemes in a way that each group has a unique pronunciation, nor do we factor in long-range context in the determination of merging or splitting. Instead, we write the rules with the expectation that the context-dependent nature of the acoustic models to learn the difference for grapheme groups having a context-dependent pronunciation. For example, we expect the acoustic models to learn from polyphone context that "ent" is voiced following an "m" and preceding a word boundary, as in "foncièrement," but that it is silent following "gn" as in "joignent." The following is a selection of some of these rules and examples of the effects of their use on the grapheme sequence of words.

é	←	{ é }, { a i }, { é e }, { u é }
ent	←	{ e n t }
e	←	{ e }, { è }, { ê }
au	←	{ a u }, { e a u }
on	←	{ o n }
oin	←	{ o i }
gn	←	{ g n }

Table 1: *Some selected rules for merging graphemes*

Because there was no prior (pseudo-)grapheme system, we trained the system using a flat-start technique based on six iterations of expectation-maximisation (EM) training and regeneration of training data alignments. When clustering quinphone models for the graphemes we used only questions about the identity of graphemes in the context of the polygraphemes, as this is known to perform quite well [15].

délaissées (*adj. fem. pl.* abandoned)	→	d é l é s é s
faisceaux (*n. m.* bundles)	→	f é s c au x
foncièrement (*adv.* fundamentally)	→	f on c i e r e m ent
joignent (*v. 3p. pl.* join)	→	j oi gn ent
pointée (*adj. fem.* pointed)	→	p oin t é

Table 2: *Selected entries from the grapheme dictionary with accompanying English translations*

3.2. Performance Comparison

The following is a comparison of the performance associated with the use of the various pronunciation models previously mentioned. The context-dependent system training is identical in every way, including feature front-end (MFCC). Only the dictionary or source of pronunciation is varied. The results are given in Table 3. The relatively higher error rate of the grapheme system relative to the phoneme systems is typical of our experience with other ASR languages [17, pg. 202].

Table 3: *Case-insensitve WER resulting from the use of various pronunciation models. The results are from systems using MFCC features and 8000 acoustic models, and are tested with the same language model.*

Dictionary	WER
BDLex	25.4
text2phone	25.6
globalphone	26.6
grapheme	27.0

4. Language Modeling

A 4-gram case-sensitive language model with modified Kneser-Ney smoothing was built for each of the text sources listed in Table 4. This was done using the SRI Language Modeling Toolkit [18]. We cleaned the Quaero acoustic training transcripts and used half as part of the training set; the other half was used as a tuning set. The language models built from these text sources were interpolated using weights estimated on this tuning set; these weights were estimated with a tool in the SRILM toolkit which uses an expectation maximization algorithm with fixed underlying mixture distributions to minimize the perplexity of the LM mixture on the tuning set. The result was a language model with 38.34 million 2-grams, 113.2 million 3-grams, and 233.8 million 4-grams.

4.1. Development

Our baseline language model was trained with newswire text from the Gigaword corpus as well as the Quaero acoustic training transcripts from all years up to 2012.

[1] From the Gigaword corpus

Source	Type	Words	Weight
Quaero transcripts	BC & BN	1M	0.459
Quaero l'Humanité	Newspaper	752M	0.127
AFP, APW, [1] and Ester	Newswire, BN	391M	0.121
Quaero Blog	Blog	62M	0.111
Quaero News Div	Newspaper	150M	0.091
AFP 2000s [1]	Newspaper	335M	0.032
CFPP2000	Interviews	417K	0.029
European parliament	Debate	100M	0.019
Est Républicain	Newspaper	104M	0.011

Table 4: *Summary of the cleaned language model (LM) training texts, including training data, type, word count per corpus, and interpolation weight in the final LM*

We achieved improvements in perplexity by including additional data in our training. In subsequent iterations of the language model, we added the Quaero 2012 additional language model training sources, among which are blog data, the newspaper *l'Humanité*, and various other news sources. We also added the transcripts of the Ester corpus [5] as well as several other sources of text we found. Among these are the newspaper *Est Républicain*, transcriptions of the European Parliament, and the small conversational corpus CFPP2000 from the University of Paris 3, composed of a collection of interviews of Parisian residents. [19].

We also reduced perplexity through normalisation of elisions, which is described in section 6.2. Last, further improvements were made by normalising the casing of our text sources using smart case models trained on large corpuses of text.

We tested the effect of these development steps by measuring the perplexity of the language model on a text set composed of several Quaero acoustic transcripts: development and evaluation 2009, evaluation 2010, and evaluation 2011. The results are shown in Table 5.

Improvement	Perplexity
Baseline	174.1
+Fast cleaned Quaero 2012 material +elision normalisation with top 50 list	153.8
+Carefully cleaned Quaero 2012 material +Ester transcripts	136.0
+Additional data sources	135.0
+Smart casing	130.3

Table 5: *Perplexity scores of successive iterations of language model development, measured on the combined 2009-2011 Quaero dev. and eval. acoustic transcripts.*

4.2. Vocabulary Selection

For selection of the search vocabulary we employed the same tuning set as used for the estimation of the LM interpolation weights. For each of the aforementioned text sources, we built a Witten-Bell smoothed unigram language model. The vocabulary of this LM was taken as the union of the vocabularies of all text sources. Using the maximum likelihood count estimation described in [20] we found the best mixture weights for representing the tuning set's vocabulary as a weighted mixture of the word counts of the sources. This gave us a ranking of all words in the union vocabulary in terms of their relevance to the tuning set. We found that a vocabulary of 250,000 words gave consistently the best performance in terms of word error rate.

5. Recognition Setup

The decoding was performed with the *Janus Recognition Toolkit* (JRTk) developed at the Karlsruhe Institute of Technology and Carnegie Mellon University [21]. Our decoding strategy is based on the combination and cross-system adaptation of many subsystems trained with different dictionaries and feature front-ends. System combination works on the principle that different systems commit different errors which cancel each other out. Cross-system adaptation profits from the fact that unsupervised acoustic model adaptation works better when performed on output that was created with a different system with comparable performance [22]. By combining subsystems with several diverse configurations, we ensure a variety of outputs upon which subsystem combination can work effectively.

5.1. Segmentation

Segmenting the input data into smaller, sentence-like chunks used for recognition was performed with the help of a fast decoding pass on the unsegmented input data in order to determine speech and non-speech regions [23]. Segmentation was then done by consecutively splitting segments at the longest non-speech region that was at least 0.3 seconds long. The resulting segments had to contain at least eight speech words and had to have a minimum duration of six seconds and a maximum length of 30 seconds.

In order to group the resulting segments into several clusters (with each cluster corresponding in the ideal case to one individual speaker) we used an hierarchical, agglomerative clustering technique based on TGMM-GLR distance measurement and the Bayesian Information Criterion (BIC) stopping criterion [24]. The resulting speaker labels were used to perform acoustic model adaptation in the multipass decoding strategy described below.

5.2. Subsystem Combination and ROVER

We use two passes of decoding. The output lattices of the subsystems in each stage are used to produce an improved output through confusion network combination (CNC) [25]. The second pass decodings are performed using SAT models and unsupervised adaptation. At this stage the subsystem makes use of the confidences of the CNC from the previous stage. As a final step we apply a ROVER for further reduc-

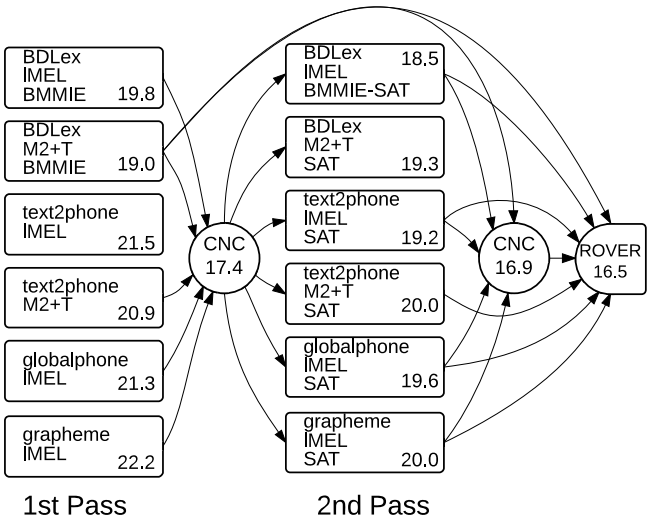


Figure 3: *Decoding strategy of the KIT 2013 evaluation system with WER for each subsystem and step.*

tion of error [26].

For each pass of CNC and ROVER we tested several combinations of subsystems in order to find the best CNC or ROVER performance on the development (2012 evaluation) set. Generally speaking, this meant leaving the one or two subsystems with the highest WER out of the CNC. However, for the ROVER we found that simply including all subsystems gave the best results. Finally, compared to the best CNC without the grapheme subsystem, inclusion of this subsystem consistently improved the CNC by approximately 0.1% absolute WER reduction. Our decoding strategy, along with WER figures for each subsystem, CNC, and the ROVER, is shown in Figure 3.

6. System Development and Experiments

In this section we describe two experiments in the development of our systems which yielded useful improvements. An overview of certain steps and their effect on the performance of the best single system is given in Table 6.

6.1. Filtering of Acoustic Training Material via Decoding

We obtained improvements in WER by undertaking a filtering of the acoustic training material. Rather than using rule-based methods, such as segment duration or relative phone duration, we used our single best system to decode on the training material. We then computed a WER for each utterance in the training database. A new training database was formed wherein those utterances scored with an error rate over a certain limit were not included. We then retrained our system on this slightly smaller, updated training database. This process was done in an iterative method with regard to the incorporation of new training material. Our acoustic training sources were divided into four sets which were each

Development step	Improvement
baseline MFCC	27.8
+ various LM improvements	26.5
+ elision normalisation with top50	26.2
+ inclusion of filtered data	25.5
+ 8K models	25.1
lMEL DBNF	19.2
M2 DBNF	19.4
M2+T DBNF	18.7
+ smart casing in LM training	18.6
lMEL DBNF + bMMIE	18.6
lMEL DBNF + SAT	17.7
lMEL DBNF + bMMIE-SAT	17.4

Table 6: *Development steps for the best single system and the resulting reduction in WER (Case Insensitive) relative to the preceding step. The test set is automatically segmented eval. 2011 data.*

filtered separately:

- "Quaero core" : Quaero partial 2010, 2011 training data
- "Quaero fast" : Quaero 2009, partial 2010 training data with "fast" transcripts
- "Quaero careful" : Quaero 2009, 2010 training data with carefully annotated transcripts
- Ester 1
- Ester 2
- Ester 2-dev

Through successive trials of experimentation we developed the following strategy. First, we adopted a rule to reject those top 10% of utterances having the highest WER. This corresponded roughly to rejecting utterances with WER $> 75\%$. As an alternative we tried rejecting the top 25% of utterances, corresponding roughly to 50% WER. Equivalent systems trained on data filtered with the 50% threshold (more strict) outperformed those trained on data filtered with the 75% threshold (less strict), as is shown in Table 7. As for the Ester data, it differs in that it is solely broadcast news and contains a good deal of telephone-quality speech. Thus we decided to apply a stricter rule of thumb that 38% WER would be the maximum for utterances from Ester.

Table 7 shows the results of successive inclusions of training material, while Table 8 shows the effects of the filtering on the amount of utilised training material.

6.2. Elision normalisation

French contains a phenomenon called elision, wherein the final vowel of one word immediately before another word beginning with a vowel is omitted and by convention the

Training material	WER
baseline (Quaero core)	27.0
baseline + filt.'d @ 75% additional (fast & careful) Quaero	26.9
baseline + filt.'d @ 50% additional (fast & careful) Quaero	26.5
baseline + filt.'d @ 50% add. Quaero + filt.'d @ 38% Ester	26.3

Table 7: *Improvements in system performance with addition of filtered data. The test set is automatically segmented eval. 2011 data. WER given is Case Insensitive.*

Training material	Unfilt. utts.	%	Unfilt. hrs.	Filt. hrs.	%
Quaero core	32.5K	100	140.0	-	100
Quaero fast	3.17K	73.4	17.5	14.8	84.76
Quaero careful	9.07K	67.9	36.7	32.9	89.6
Ester 1	11.6K	48.5	61.6	48.3	78.3
Ester 2	6.58K	50.4	40.6	28.2	69.6
Ester 2 dev	1.32K	48.6	5.65	3.76	66.7
Total			302	268	88.7

Table 8: *Effects of decoding-based filtering on training material.*

two words are joined with an apostrophe in the written form. For example "ce est" becomes "c'est" ("this is,") "la apparance," becomes "l'apparance" ("the appearance,") "de être" becomes "d'être," ("to be") and so on. When we consider these joined words to be one unit for language modeling (a treatment we call *"join"*) we are faced with a challenge due to a large number of out-of-vocabulary (OOV) words which are simply the elision of two words already in our search vocabulary. The natural solution would then seem to be to consider these units as two separate words (a treatment we call *"separate"* or *"sep."*) While this reduces OOV frequencies, it decreases the language model context. We took a compromise approach by treating the fifty most common elided word combinations in our training transcripts as one word and the rest as two (a treatment we call *"sep. w/ top50."*) This was reflected in our text normalisation.

We selected three 250,000-word search vocabularies from the same data, appropriately filtered in each case to treat elision differenty according to the schemes described above. Table 9 shows the OOV rate of these vocabularies as tested on the 2011 evaluation Quaero transcripts (appropriately processed to reflect the same treatment of elisions.) We see that treating elided combinations as two separate words dramatically reduces OOV. As a further experiment, we trained three otherwise identical recognisers, each reflecting one of these treatments of elision and tested them with a corresponding language model. The effects are shown in the same table (Table 9). The separated approach outperfoms the joined approach, and joining the fifty most common elisions gives further gains.

7. Conclusions

In this paper we presented the KIT French Quaero Speech-to-Text system. We described details of its development through the integration of enhancements such as deep neural networks for features, tonal features, multiple pronunciation models including a modified grapheme scheme, and other development techniques used to improve recognition accuracy. The combination of these techniques was shown to significantly reduce error on this task. Future system development should focus methods on advanced language modeling techniques such as neural networks and techniques developed for inflectional languages in an effort to reduce homophone confusability. Further refinements may also be made to our acoustic neural networks by using multilingual training data and training from new alignments written with neural network systems.

Elision treatment	S. Vocab oov	WER
join	5.0%	29.6%
sep.	0.61%	26.7%
sep. w/ top50	0.68%	26.2%

Table 9: *OOV rate for 250K-word vocabularies on eval2011 data.*

8. Acknowledgements

This work was realised as part of the Quaero Programme, funded by OSEO, the French state agency for innovation.

9. References

[1] M. Gales, "Semi-tied covariance matrices for hidden markov models," Cambridge University, Engineering Department, Tech. Rep., February 1998.

[2] T. Anastasakos, J. McDonough, and J. Makhoul, "Speaker adaptive training: A maximum likelihood approach to speaker normalization," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2. IEEE, 1997, pp. 1043–1046.

[3] M. J. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[4] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted mmi for model and feature-space discriminative training," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4057–4060.

[5] S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.-F. Bonastre, and G. Gravier, "The ester phase ii evaluation campaign for the rich transcription of french broadcast news." in *Interspeech*, 2005, pp. 1149–1152.

[6] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-

encoders," in *ICASSP2013*, Vancouver, CA, 2013, pp. 3377–3381.

[7] Q. B. Nguyen, J. Gehring, K. Kilgour, and A. Waibel, "Optimizing deep bottleneck feature extraction," in *RIVF2013*, Hanoi, Vietnam, 2013.

[8] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec. 2010.

[9] F. Metze, Z. Sheik, A. Waibel, J. Gehring, K. Kilgour, Q. Nguyen, and V. Nguyen, "Models of tone for tonal and non-tonal languages," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2013, to appear.

[10] K. Schubert, "Grundfrequenzverfolgung und deren anwendung in der spracherkennung," master's thesis, Universität Karlsruhe (TH), Germany, 1999, in German.

[11] K. Laskowski, M. Heldner, and J. Edlund, "The fundamental frequency variation spectrum," *Proceedings of the Swedish Phonetic Conference (FONETIK 2008)*, pp. 29–32, June 2008.

[12] D. Haubensack, "Text2phone."

[13] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[14] C. Schillo, G. A. Fink, and F. Kummert, "Grapheme based speech recognition for large vocabularies," in *Proceedings of the Sixth International Conference on Spoken Language Processing (ICSLP 2000)*. Beijing, China: ISCA, October 2000, pp. 584–587.

[15] M. Killer, S. Stüker, and T. Schultz, "Grapheme based speech recognition," in *Proceedings of the 8th European Conference on Speech Communication and Technology EUROSPEECH'03*. Geneva, Switzerland: ISCA, September 2003, pp. 3141–3144.

[16] S. Kanthak and H. Ney, "Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition," in *Proceedings the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, vol. 1. Orlando, Florida, USA: IEEE, 2002, pp. 845–848.

[17] K. Kilgour, C. Saam, C. Mohr, S. Stüker, and A. Waibel, "The 2011 kit quaero speech-to-text system for spanish," *IWSLT-2011*, pp. 199–205, 2011.

[18] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *ICSLP*, 2002.

[19] S. Branca-Rosoff, S. Fleury, F. Lefeuvre, and M. Pires, "Discours sur la ville," *Corpus de français parlé parisien des années*, 2000.

[20] A. Venkataraman and W. Wang, "Techniques for effective vocabulary selection," *Arxiv preprint cs/0306022*, 2003.

[21] H. Soltau, F. Metze, C. Fuegen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *ASRU*, 2001.

[22] S. Stüker, C. Fügen, S. Burger, and M. Wölfel, "Cross-system adaptation and combination for continuous speech recognition: the influence of phoneme set and acoustic front-end." in *INTERSPEECH*, 2006.

[23] S. Stüker, C. Fügen, F. Kraft, and M. Wölfel, "The isl 2007 english speech transcription system for european parliament speeches." in *INTERSPEECH*, 2007, pp. 2609–2612.

[24] Q. Jin and T. Schultz, "Speaker segmentation and clustering in meetings," in *Proceedings of the 8th International Conference on Spoken Language Processing (Interspeech 2004 — ICSLP)*. Jeju Island, Korea: ISCA, October 2004.

[25] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, October 2000.

[26] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proceedings the IEEE Workshop on Automatic Speech Recognition and Understanding*. Santa Barbara, CA, USA: IEEE, December 1997, pp. 347–354.

Improving Bilingual Sub-sentential Alignment by Sampling-based Transpotting

Li Gong, Aurélien Max, François Yvon

LIMSI-CNRS & Univ. Paris Sud
Orsay, France
{firstname.lastename}@limsi.fr

Abstract

In this article, we present a sampling-based approach to improve bilingual sub-sentential alignment in parallel corpora. This approach can be used to align parallel sentences on an *as needed* basis, and is able to accurately align newly available sentences. We evaluate the resulting alignments on several Machine Translation tasks. Results show that for the tasks considered here, our approach performs on par with the state-of-the-art statistical alignment pipeline `giza++/Moses`, and obtains superior results in a number of configurations, notably when aligning additional parallel sentence pairs carefully selected to match the test input.

1. Introduction

Sub-sentential alignment consists in identifying translation units from a sentence-aligned parallel corpus, which is a crucial component of state-of-the-art Statistical Machine Translation (SMT) technology. One of the most prominent approaches nowadays is Phrase-based Statistical Machine Translation, which is built upon the word alignment output. The problem of learning sub-sentential alignment from parallel texts is well-known, and numerous proposals have been put forward to perform this task. Those methods roughly fall into two main categories, broadly described here as the *probabilistic* and the *associative* approaches.

The probabilistic approach, introduced in [1], considers the problems of identifying *links* between words or groups of words in parallel sentences. This approach consists in defining a probabilistic model (e.g. IBM models [2]) of the parallel corpus, the parameters of which are estimated by a global optimization process which simultaneously considers all possible associations in the entire corpus. Due to its tight integration within the SMT framework, this approach is by far the most widely used. However, it is characterized by a number of shortcomings, in particular:

- Its parameters have to be estimated and optimized based on the entire parallel corpus, hence all units in the parallel corpus have to be aligned simultaneously. This makes it a time-consuming process, especially when working on large parallel corpora. In addition, many aligned parallel sentence pairs are never used to translate an input text.
- New data are constantly made available. It is a waste of resource to run the alignment process repeatedly for the whole corpus when only a proportionally low number of new sentences are added.

These shortcomings are addressed notably in [3], which uses the online EM algorithm of [4] to implement online learning for the HMM alignment model.

Associative approaches were introduced in [5]. They do not rely on an alignment model, but rather on independence statistical measures such as the Dice coefficient, mutual information [5, 6], or likelihood ratio [7]. In this approach, a local maximization process is used, where each sentence is processed independently.

An associative sub-sentential alignment method, named `Anymalign`, was introduced in [8, 9]. This method relies on simple comparisons on (source and target) word occurrence distribution over randomly sampled sub-corpora. Words with the same occurrence distribution over a particular sub-corpus are extracted as an association. The more often two words are associated, the better the association score between them, and the more likely they are to be mutual translations. This method was shown to produce better results than state-of-the-art methods on bilingual lexicon constitution tasks, when the evaluation is performed by comparing word associations with reference dictionaries, but failed to perform on par with state-of-the-art methods for building SMT phrase tables. It was subsequently improved in [10], in which a recursive binary segmentation algorithm is used to process the output of `Anymalign` so as to obtain better sub-sentential alignments at the sentence level. While this improvement yields a performance that is comparable with the statistical approach, it can do so by processing large numbers of randomly sampled sub-corpora in order to obtain an accurate association measure and a good coverage for the entire corpus.

In this work, we propose a method to adapt `Anymalign` in order to align the parallel sentences on a per-need basis, meaning that it can also be used to accurately align new parallel sentences as they become available. The rest of this paper is organized as follows: Section 2 describes our sampling-based alignment approach in some detail, Section 3 presents an evaluation on several, complementary Machine Translation experiments, and Section 4 discusses our main results and introduces some of our future work.

2. Description of the method

We assume that, given a parallel bilingual corpus C, we wish to align several sentence pairs in a set S: S can be a part of the entire parallel bilingual corpus ($S \subseteq C$), or can correspond to newly available data ($S \nsubseteq C$).

An association table is first extracted for sentences in S by a *sampling-based transpotting* method. This table contains only the source phrases that exist in some sentence(s) of S. Using this table, a recursive binary segmentation algorithm (as in [10]) is applied to each sentence pair of S so as to generate the desired sub-sentential alignment.

2.1. Sampling-based transpotting

Our sampling-based transpotting method is inspired by `Anymalign`, which aims at extracting sub-sentential associations from multilingual, parallel corpora. `Anymalign` repeatedly draws random sub-corpora from the full parallel corpus, and extracts associations from each sub-corpora, which are used to build an association table between phrases. As each sub-corpora is independent, this process could be stopped at any time. However, large numbers of sub-corpora have to be processed in order to achieve a good coverage of the phrases in the entire corpus.

In our work, `Anymalign` is adapted in order to extract an association table for a specific list of sentence pairs S. Each sentence pair $(\mathbf{s}, \mathbf{t})$ in S is processed separately and a number N of random sub-corpora are sampled from the full parallel corpus C for each sentence. For each sub-corpora, the distribution profile is computed only for words (or phrases) occurring in s and bilingual phrases with the same profile are extracted as likely associations. The more sub-corpora are processed for each sentence pair, the more associations could be extracted, and the more accurate the association measures are. The set of all associations extracted from each sentence pair form the association table of S. In a nutshell, this procedure performs bi-sentence alignment *via* transpotting based on randomly sampled sub-corpora. The complete process is illustrated on an English-French sentence pair on Figure 1.

There are notable differences between this method and `Anymalign`:

- `Anymalign` draws random sub-corpora from the parallel corpus, and computes the occurrence distribution profile for all words of all sentence pairs in the sub-corpora, while we need to compute such profiles only for words in the sentence pair to align.[1]
- `Anymalign` is *anytime* but typically requires a large number of sub-corpora to achieve a good coverage over the entire corpus. We draw N sub-corpora for each given sentence pairs to ensure better coverage for the contents of each sentence pair to align. This allows

[1] Note that, when one's objective is in fact to align a complete parallel corpus, all counts should be kept.

(1) Given a source-target sentence pair, we need to extract an association table for it:

one coke , please . ↔ un coca, s'il vous plaît .

⇓

(2) Draw a random sub-corpus from the parallel corpus:

	English	*French*
1	*one coffee, please .*	*un café, s'il vous plaît .*
2	*the coffee is not bad .*	*ce café est correct .*
3	*yes, one tea .*	*oui, un thé .*

⇓

(3) Compute occurrence distribution profile for words in the current sentence pair:

words with same distribution profile			*profiles*
one ,	↔	*un ,*	*[1, 0, 1]*
coke	↔	*coca*	*[0, 0, 0]*
please	↔	*s'il vous plaît*	*[1, 0, 0]*
.	↔	*.*	*[1, 1, 1]*

⇓

(4) If the source and target phrases are each contiguous, then increment the count for the corresponding phrase pair:

1. *count of (coke↔coca) plus 1*
2. *count of (please↔s'il vous plaît) plus 1*
3. *count of (. ↔ .) plus 1*

⇓

(5) Repeat steps (2) and (4) N times, so as to obtain an association table for the given sentence pair, e.g.:

source phrase		*target phrase*	*count*
one	↔	*un*	*830*
coke	↔	*coca*	*560*
one coke	↔	*un coca*	*20*
,	↔	*,*	*900*
please	↔	*s'il vous plaît*	*160*
please	↔	*s'il*	*200*
please	↔	*plaît*	*500*
.	↔	*.*	*980*

Figure 1: Illustration of the sampling-based transpotting method on an English-French sentence pair.

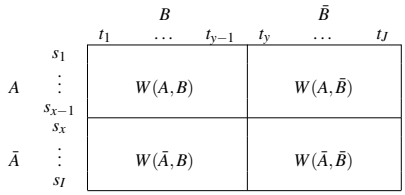


Figure 2: Schematic representation of the segmentation of a pair of sentences $S = A.\bar{A}$ and $T = B.\bar{B}$ (from [10]).

to align sentences on a per-need basis, and furthermore offers a more interpretable running time, which is now controlled by the amount of desired sampling for each sentence pair, which could e.g. depend on its length.

2.2. Sub-sentential alignment extraction

Once the association table for some sentence pairs is obtained, a recursive binary segmentation algorithm, described in [10], and inspired by the work of [11, 12], is used to generate a sub-sentential alignment for each sentence pair. Its purpose is to recursively segment the source and target sentence simultaneously on the basis of local association scores so as to find the *links* between the source and target words. It thus requires some association score $w(s,t)$ between each source word s and target word t in a sentence pair, which can be the result of the process described in Section 2.1. Then, recursive binary segmentation is guided by the sum W of the association scores between each source and target words of a block $(X,Y) \in \{A,\bar{A}\} \times \{B,\bar{B}\}$ (as shown in Figure 2):

$$W(X,Y) = \sum_{s \in X, t \in Y} w(s,t) \tag{1}$$

The best segmentation is the one which minimizes the score defined in Equation 2:

$$\text{cut}(X,Y) = W(X,\bar{Y}) + W(\bar{X},Y) \tag{2}$$

which would indicate that the association between the words of X and $\bar{Y}$ on the one hand, and the words of $\bar{X}$ and Y on the other hand, have low association scores. Following [10], we use instead a normalized variant so as to not to encourage unbalanced segments:

$$\text{Ncut}(X,Y) = \frac{\text{cut}(X,Y)}{\text{cut}(X,Y)+2\times W(X,Y)} + \frac{\text{cut}(\bar{X},\bar{Y})}{\text{cut}(\bar{X},\bar{Y})+2\times W(\bar{X},\bar{Y})} \tag{3}$$

With this segmentation criterion, the binary segmentation algorithm tests every possible binary segmentation in order to find the best segmentation score, and recursively segments blocks in a greedy fashion. In our current implementation, the segmentation terminates on blocks with at least one side of length 1 token. Figure 3 shows an example of segmentation, where atomic aligned biphrases correspond to framed rectangles containing values in bold. The words in aligned biphrases are linked with each other, which forms the word-to-word alignment of the bisentence.

	un	coca	,	s'il	vous	plaît	.
one	**0.246**	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
coke	ϵ	**0.138**	ϵ	ϵ	ϵ	ϵ	ϵ
,	ϵ	ϵ	**0.624**	0.002	ϵ	ϵ	0.048
please	ϵ	ϵ	ϵ	**0.032**	**0.008**	**0.128**	ϵ
.	ϵ	ϵ	0.020	ϵ	ϵ	ϵ	**0.873**

Figure 3: Example of alignment by recursive segmentation. The number in each cell corresponds to the value of function w, with $0 < \epsilon \leq 0.001$.

2.3. Self-convergency normalization

Segmentation scores for each position of token pairs are initialized by looking up values in the association table obtained by sampling-based transpotting (see Section 2.1). Because these association scores may sometimes be unreliable and poor indicators of a translation relationship, the best-first segmentation algorithm may produce incorrect results, especially on long sentence pairs. In addition, the bilingual sentence pairs are often in some relation to each other. So, well aligned sentences can help improve the alignment of more difficult sentences.

Therefore, we propose to use the previously produced alignment to extract the source-target phrase pairs to build an updated association table. This new table can then be used for another, better informed pass of recursive segmentation. This can be repeated until the obtained alignments are stable across iterations. This is described in Algorithm 1, where distance$(A - A')$ is the percentage of different links between A and A'.

Algorithm 1 Self-convergency normalization

Given a parallel corpus C and its alignment A
NumIter=0
while NumIter $<$ MaxIter **do**
  Extract all aligned source-target phrases from C using A with the same heuristic as `Moses`
  The extracted phrase pairs and their counts are used to build an association table T (the same kind of table as the table in step 5 in Figure 1)
  Using T as the input of the binary segmentation algorithm (cf. Section 2.2), a new alignment A' is computed
  if distance$(A - A') < \epsilon$ **then**
    return A
  end if
  NumIter+=1
end while
return A

3. Experiments

3.1. Experimental settings

In this section, we describe experiments intended to test the performance of the associative sub-sentential alignment ap-

proach described in Section 2. We will focus on measuring the impact of several alignment strategies for a phrase-based SMT system. We will use the Moses toolkit [13], which can be regarded as state-of-the-art for building SMT systems. Moses will be used in all configurations to build phrase tables and reordering tables from alignment matrices, and its decoder will be used to build candidate translations during optimization (using standard MERT [14]) and testing.

Translation performance will be measured by classical corpus-based metrics, BLEU [15] and TER [16]. All results are average scores computed on the test set for 3 independent optimization runs on the development set [17].

Experiments will be conducted on three language pairs and two main corpora, and we will make use of several reference translations when possible. We will also resort to oracle decoding using a greedy, approximate local search strategy and a number of phrase-based operators [18] to get some account of the best translation score attainable given each specific phrase table. We will furthermore consider the compactness of the produced phrase tables, as it can be regarded as a desirable quality of phrase tables licencing works on phrase table pruning (see e.g. [19]), and anormally large phrase table may in fact only artificially inflate oracle results.

Two sets of experiments will be carried out in this work. The first set of experiments is designed to validate the quality of the alignment generated by our method (henceforth sba, for sampling-based alignment) on some predefined bilingual corpus against a state-of-art alignment pipeline, based on giza++ [20], using default parameters from Moses. This approach is refered to as giza++ . The second set of experiments aims to assess the ability to align new bilingual data. For this experiment, we will focus on adding sentence pairs from a very large (unaligned) bilingual corpus, chosen on the basis that they contain translations for previously out-of-vocabulary tokens. Our approach will be compared against the same alignment pipeline using the augmented parallel corpus. This strategy is however costly as it requires to retrain the complete models, so we also performed a comparison with alignments obtained using the orginal alignment models, without any retraining.

3.2. Data sets

Experiments were performed on two parallel corpora, described in Table 1: BTEC is a small English-French subpart of the Basic Travel Expression Corpus [21]; and HIT is a corpus of basic expressions built for the Beijing 2008 Olympics, used here in English, French and Chinese. We used the BTEC development set of 2003 (devel03) and BTEC test set of 2009 (test09) as our development and test set, which are described in Table 2. Note that the former has 16 reference translations available for English, and the latter has 7, allowing for a somehow more interpretable measure of performance for language pairs with English as the target language.

We will describe in Section 3.4 experiments that make use of additional data extracted from the large EPPS (Europarl) English-French parallel corpus of parliamentary debates, as well as a substantially larger corpus from the translation task of the Workshop on Statistical Machine Translation (WMT)[2]: both are described in Table 1. Our development and test sets will remain the same for all experiments.

Corpus	# lines	#token$_{en}$	# token$_{fr}$	# token$_{zh}$
BTEC	20K	182K	207K	-
HIT	62K	600K	690K	590K
EPPS	1,982K	54,170K	59,702K	-
supp	3.3K	111K	121K	-
WMT	11,745K	317,688K	383,076K	-

Table 1: Training bitext corpora statistics

Corpus	#lines	Avg(#token$_{en}$)	#token$_{fr}$	#token$_{zh}$
devel03	506	4,098 (16 refs)	4,220	3,435
test09	469	3,928 (7 refs)	4,023	3,031

Table 2: Tuning and test sets statistics

English and French texts are normalized and tokenized by our in-house tools, and Chinese texts are segmented by a CRF-based Chinese word segmenter[3].

3.3. Basic alignment task

This experiment aims to assess the quality of the subsentential alignment generated by our method on a full bilingual parallel corpus. We use the giza++ implementation of [22] as a competitive baseline, with default settings : 5 iterations of IBM1, HMM, IBM3, and IBM4, in both directions (source to target and target to source). As for our alignment method, its alignment quality depends on the number of sub-corpora (N) that are drawn for each sentence pair. In this work, we choose a constant value of $N = 1000$ for all sentence pairs. The self-convergency normalization process is repeated for a maximum of 10 iterations.

The results for the two alignment methods are reported in Table 3, where we compare them on 2 parallel corpus (**BTEC** and **HIT**) and their simple concatenation (**BTEC+HIT**) and 3 translation directions on the same test set.

3.3.1. In-domain evaluation

First, on the in-domain corpus, **BTEC**, we find that our approach performs better than giza++, in particular by a large margin on the single-reference English→French direction (average of +2.13 BLEU). These results are furthermore obtained using a substantially smaller phrase table (315K vs. 360K entries in the phrase tables). Oracle-BLEU also indicates a clear advantage for our approach (average

[2] http://www.statmt.org/wmt12

[3] http://nlp.stanford.edu/software/segmenter.shtml

	BTEC				HIT				BTEC+HIT			
	BLEU	oracle-BLEU	TER	# entries	BLEU	oracle-BLEU	TER	# entries	BLEU	oracle-BLEU	TER	# entries
English→French (1 reference)												
giza++	45.68	76.26	37.03	360K	39.65	68.20	44.50	1,217K	47.97	83.62	35.45	1,546K
sba	47.81	77.78	36.60	315K	39.70	68.45	43.56	921K	47.55	84.40	37.22	1,241K
French→English (7 references)												
giza++	59.50	77.23	24.59	360K	45.52	68.58	33.99	1,224K	63.69	84.00	21.95	1,551K
sba	59.92	77.50	24.22	315K	45.34	69.59	33.79	937K	64.44	83.57	22.31	1,241K
Chinese→English (7 references)												
giza++	-	-	-	-	27.88	51.69	50.76	1,139K	-	-	-	-
sba	-	-	-	-	27.85	53.05	50.93	655K	-	-	-	-

Table 3: Results of experiments where specific bilingual parallel corpora are fully aligned. Values all correspond to average scores over three decodings of the test file for 3 independent optimization runs.

of +1.52 BLEU). These last two results are possible indicators of the fact that our approach produced a better subsentential alignment of the parallel corpus: better results can be (and are) obtained although fewer phrase pairs were extracted from the corpus.

3.3.2. Multiple-reference evaluation

Looking at the opposite translation direction with 7 reference translations, French→English, we still find that our technique is superior to the baseline, although to a much more modest extent (averages of +0.42 BLEU for the one-best translation and +0.27 BLEU for the oracle). Using several reference translations can potentially help us ensure that measured improvements are more related to *actual improvements* that e.g. make translation lexically more appropriate, than to specific choices that would accidentally resemble some particular reference translation. Again, our three indicators (one-best translation, oracle translation, and phrase table size) all indicate that our approach is here superior to the baseline.

3.3.3. Out-of-domain evaluation

Moving to the slightly less in-domain **HIT** corpus (the baseline performance drops from 59.50 to 45.52 BLEU on French→English), we find that the two approaches now perform roughly in the same ballpark, with our approach still producing significantly more compact phrase tables. For the more interpretable French→English condition with 7 reference translations, we find that although BLEU cannot be used to decide between the two, the oracle value still indicates a large advantage for our sampling-based alignment (average of +1.01 BLEU). This means that it managed to extract more useful phrase pairs, but that their various scores could not be used to ensure that those would be used in the one-best hypotheses of the decoder. Given that **HIT** is of a different origin than the test corpus (**BTEC**), it is well conceivable that translation preferences or even senses can often differ, resulting in some appropriate translation hypotheses with low scores that prevent them from appearing in one-best hypotheses.

3.3.4. Larger, composite training corpus evaluation

The previous hypothesis seems to hold when considering the larger task corresponding to the concatenation of the two parallel corpora (**BTEC+HIT**), where **HIT** data outnumber **BTEC** data by more than 3:1. Results are however less clear-cut here: for instance, our approach still performs better on French→English (average of +0.75 BLEU on one-best hypotheses), but fares worse in terms of oracle performance (average of -0.43 BLEU). These results include a reflection of the fact that `giza++` improves its alignment with more data, even when adding out-of-domain data [23]. At this stage of our work, we do not control which particular sentence pairs are drawn in our samples, so assessing the impact of a larger overall sentence pool cannot be done.

3.3.5. Difficult language pair evaluation

Lastly, we turn to the more difficult Chinese→English condition, which is significantly more difficult than its French→English counterpart (27.88 BLEU vs. 45.52 BLEU for the `giza++` baselines). A similar pattern emerges for the two language pairs: one-best translation performance is comparable, but oracle results indicate a clear advantage for our sampling-based alignment (average of +1.36 BLEU). Furthermore, for this language pair, we find that this is obtained with significantly fewer phrase table entries (almost half as many). Chinese words may in fact be very difficult to align to English words, partly for ambiguity reasons, and many noisy translation candidates may be extracted. Additionally, many words may be left unaligned by `giza++`, leading to artificially large numbers of extracted phrase pairs by the default `grow-diag-final-and` heuristic.

3.4. Incremental alignment task

In the previous section, we have shown that our approach performs on par with the `giza++` baseline on the studied configurations for full corpus alignment. We now turn to the issue of aligning new data, which in many situations could

Phrase tables				HIT					
main	**supplementary**								
(62K HIT)	(3.3K EPPS)	# entries	# transl.	BLEU	1g	2g	3g	4g	TER
French→English (7 references)									
giza++	none	-	-	45.52	76.5	52.2	37.8	27.1	33.99
|	forced	59	1,993	47.94	76.8	55.4	41.0	29.2	34.62
|	concat	60	1,190	48.69	78.4	56.1	41.4	29.8	33.09
|	sba	64	681	49.83	80.9	57.3	42.0	30.5	30.61
|	concat++	62	1,218	50.23	81.5	57.8	42.6	31.1	29.81
sba	none	-	-	45.34	77.0	52.1	37.4	26.9	33.79
|	sba	64	681	50.45	81.8	58.3	42.5	30.9	29.94

Table 4: Results of experiments where a supplementary corpus is pooled and aligned by several methods.

only be performed on demand. Indeed, considering that all input sentences in our test set could be translated independently at large intervals of time, it would certainly not be conceivable, time-wise and computation-wise, to perform a full statistical alignment of the iteratively growing bilingual corpus. We will nonetheless report evaluation results for this situation below.

Few works have previously considered the task of incremental alignment of parallel corpora [24, 25]. The focus in [25] is put on a careful selection of additional data, a reflection of the fact that not all training data can be beneficial for training and improving SMT systems [26]. For these experiments, we will concentrate on a very specific use of additional data with a conservative view[4]: sentences will be pooled from a very large, any-domain parallel corpus (EPPS in Table 1) on the basis that they contain at least one occurrence of a word that is out-of-vocabulary (OOV) in the baseline parallel corpus[5]. In order to study a condition where significant numbers of such OOVs exist, we used the **HIT** corpus as our main corpus, relatively to which our test set contains 79 unique OOVs (436 occurrences). Our additional training data (**EPPS**) provided matches for 65 of them. We retrieved a maximum of 100 sentences pairs for each of these 65 OOVs, which yielded an additional parallel corpus of 3,355 sentence pairs (**supp** in Table 1).

We now describe the configurations that will be compared. A main table will be used for all configurations, corresponding either to the giza++ baseline or to our sampling-based approach. A supplementary table will be built from **supp** by various means:

- forced alignment on **supp** using the statistical models (previously) obtained on **HIT** (forced);
- statistical alignment on the concatenation **HIT+supp**, and extraction of the alignments on **supp** only (concat);
- sampling-based alignment on **supp**, sampling from the union of **HIT** and **supp** (sba);
- statistical alignment on the very large corpus used for experiments at WMT'12 [27], and extraction of the alignments on **supp** only (concat++).

As said previously, the concat variants cannot be considered as practical solutions for the problem at hand. Once alignments are obtained for the **supp** corpus, a separate phrase table is used by the Moses tools as previously, and MERT is used with the resulting two tables, where our additional table is used as backoff, for unigrams only. Therefore, our additional training data, once aligned, will only be used in practice for proposing translations for previously unknown words. Note that in this experiment we do not extract necessary information to update the lexicalized reordering models used by Moses.

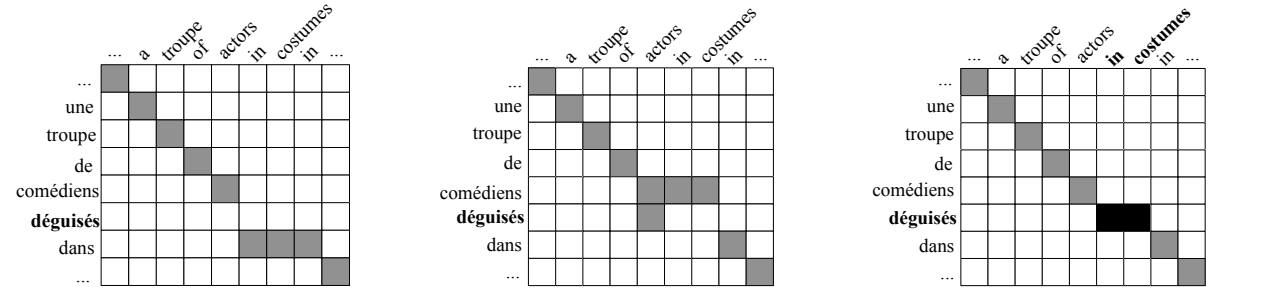
Results for this set of experiments are given in Table 4. Using giza++ for building the main translation table, we find a very clear ranking for all the studied strategies: concat++ > sba > concat > forced > none. The only approach that outperforms ours (average of +0.4 BLEU) is the statistical alignment technique using more than 11.7M sentence pairs[6]. sba outperforms concat (average of +1.14 BLEU) and forced (average of +1.89 BLEU), the latter being the most practical baseline to consider. Significant improvements can be observed on 1-gram precision, which percolate nicely to higher-order n-grams. We note once more that our technique produces much smaller phrase tables, and further note that the concat variants already significantly reduce the numerous entries produced by forced.

Interestingly, we manage to improve this result further by using also our sampling-based alignment technique for aligning the main parallel corpus (average of +0.62 BLEU), which furthermore happens to be even slightly superior to concat++ (average of +0.22 BLEU, with small improvements on 1-gram and 2-gram precisions). To explain this fact, we return to our oracle results reported in Table 3 on

[4]We, however, do not have the guarantee that even if translations are correctly extracted, those will be those found in the reference translations.

[5]Meaning that the word was not present in the original training data, not that no translation for it could be extracted by some technique.

[6]This alignment process took roughly 2 weeks using modern computing resources.



(a) giza++ forced alignment (b) giza++ concat (c) sampling-based alignment

Figure 4: Example of matrices on French-English obtained using two giza++ baselines and our sampling-based strategy.

HIT for French-to-English translation. We there found that one-best translation was slightly superior for the baseline (average of -0.18 BLEU), but that the oracle for our approach was superior (average of +1.01 BLEU), indicating that our approach did extract more useful phrases, but which were apparently poorly scored, possibly due to domain mismatch between training and testing. It seems that providing the decoder with translation for previously OOV words had an additional effect on the configuration where we use the phrase table obtained using our technique: such translations now seem to be selected more often, resulting e.g. in a largely improved 1-gram precision by using our additional phrase table (+4.8).

4. Discussion and future work

In this work, we have presented an extension of the work by [10] on sampling-based alignment and a number of experiments that have shown its very competitive performance. Our approach performed at worse on par with a state-of-the-art baseline implementing a probabilistic approach, and obtained superior results in a number of configurations. Its more apparent strength emerged when aligning new data containing highly useful words (words that were previously out-of-vocabulary in the available data). While it remains to be shown more formally, we hypothesize that these improvements mainly stem from the improved alignment of rare words and its cascading effects. Figure 4 illustrates a case where the rare French word *déguisés* (here: *in costumes*) was only correctly aligned by our technique, and where the negative consequences for the two giza/moses baselines could be important (at least, for our experiments, no translation for *déguisés* alone could be extracted from this sentence pair by giza++ here).

The framework that we have described for targeted additional data selection from parallel corpora will be the basis for our future work. We can, by principle, work at the level of tera-scale translation [28], by accessing efficiently (using suffix arrays) large quantities of unaligned parallel corpora, and perform transpotting and phrase table construction on a per-need basis. However, considering the diversity in nature, origin and quality of all possibly additional training examples, some adaptation should be performed so as to introduce preferences for the most promising examples, and hence extracted translations. In this context, the most realistic scenario will be a follow-up to our previous work on *any-text* translation [29], where notably little or no a priori knowledge exists about (additional) training examples, and adaptation should be performed on-the-fly. Finally, it seems obvious that the search for new translations, and in particular for unknown words and phrases as well as poorly adapted phrases, should also be pursued in *less parallel* corpora (see e.g. [30]). It is then an interesting question to consider how our technique would fare and how it could be adapted to work indifferently on parallel or reasonably comparable sentence pairs.

5. Acknowledgements

This work was partially funded by the French State agency for innovation (OSEO) in the Quaero Programme.

6. References

[1] P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, and P. Roossin, "A statistical approach to language translation," in *Proceedings of COLING*, 1988, pp. 71–76.

[2] P. Brown, V. Della Pietra, S. Della Pietra, and R. Mercer, "The mathematics of Statistical Machine Translation: parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, 1993.

[3] A. Levenberg, C. Callison-Burch, and M. Osborne, "Stream-based translation models for Statistical Machine Translation," in *HLT: The 2010 Annual Conference of NAACL*, 2010, pp. 394–402.

[4] O. Cappé and E. Moulines, "On-line expectation–maximization algorithm for latent data models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 3, pp. 593–613, 2009.

[5] W. A. Gale and K. W. Church, "Identifying word correspondence in parallel texts," in *Proceedings of the Workshop on Speech and Natural Language*, Pacific Grove, USA, 1991, pp. 152–157.

[6] P. Fung and K. W. Church, "K-vec: A new approach for aligning parallel texts," in *Proceedings of COLING*, Kyoto, Japan, 1994, pp. 1096–1102.

[7] T. Dunning, "Accurate methods for the statistics of surprise and coincidence," *Computational linguistics*, vol. 19, no. 1, pp. 61–74, 1993.

[8] A. Lardilleux and Y. Lepage, "Sampling-based multilingual alignment," in *Proceedings of RANLP*, Borovets, Bulgaria, 2009, pp. 214–218.

[9] A. Lardilleux, F. Yvon, and Y. Lepage, "Generalizing sampling-based multilingual alignment," *Machine Translation*, vol. 27, no. 1, pp. 1–23, 2013.

[10] ——, "Hierarchical Sub-sentential Alignment with Anymalign," in *Proceedings of EAMT*, Trento, Italy, 2012, pp. 280–286.

[11] D. Wu, "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora," *Computational linguistics*, vol. 23, no. 3, pp. 377–404, 1997.

[12] Y. Deng, S. Kumar, and W. Byrne, "Segmentation and alignment of parallel text for statistical machine translation," *Natural Language Engineering*, vol. 13, no. 03, pp. 235–260, 2006.

[13] P. Koehn, A. Birch, C. Callison-burch, M. Federico, N. Bertoldi, B. Cowan, C. Moran, C. Dyer, A. Constantin, and E. Herbst, "Moses : Open Source Toolkit for Statistical Machine Translation," in *ACL, demo session*, Prague, Czech Republic, 2007, pp. 177–180.

[14] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proceedings of ACL*, Sapporo, Japan, 2003, pp. 160–167.

[15] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of ACL*, 2002, pp. 311–318.

[16] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of AMTA*, Cambridge, USA, 2006, pp. 223–231.

[17] J. H. Clark, C. Dyer, A. Lavie, and N. A. Smith, "Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability," in *Proceedings of ACL*, Portland, USA, 2011, pp. 176–181.

[18] B. Marie and A. Max, "A Study in Greedy Oracle Improvement of Translation Hypotheses," in *IWSLT, Heidelberg, Germany*, 2013.

[19] R. Zens, D. Stanton, and P. Xu, "A Systematic Comparison of Phrase Table Pruning Techniques," in *Proceedings of EMNLP*, Jeju Island, Korea, 2012, pp. 972–983.

[20] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[21] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, "Toward a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversations in the Real World," in *Proceedings of LREC*, Las Palmas, Spain, 2002.

[22] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Software Engineering, Testing, and Quality Assurance for NLP*, 2008, pp. 49–57.

[23] K. Duh, K. Sudoh, and H. Tsukada, "Analysis of translation model adaptation in Statistical Machine Translation," in *Proceedings of IWSLT*, Paris, France, 2010.

[24] Q. Gao, W. Lewis, C. Quirk, and M.-Y. Hwang, "Incremental Training and Intentional Over-fitting of Word Alignment," in *Proceedings of MT Summit*, Xiamen, China, 2011.

[25] P. Banerjee, S. K. Naskar, J. Roturier, A. Way, and J. van Genabith, "Translation Quality-Based Supplementary Data Selection by Incremental Update of Translation Models," in *Proceedings of COLING*, Mumbai, India, 2012, pp. 149–166.

[26] G. Gascó, M.-A. Rocha, G. Sanchis-Trilles, J. Andrés-Ferrer, and F. Casacuberta, "Does more data always yield better translations?" in *Proceedings of EACL*, Avignon, France, 2012, pp. 152–161.

[27] H.-S. Le, T. Lavergne, A. Allauzen, M. Apidianaki, L. Gong, A. Max, A. Sokolov, G. Wisniewski, and F. Yvon, "LIMSI @ WMT12," in *Proceedings of WMT*, Montréal, Canada, 2012, pp. 330–337.

[28] A. Lopez, "Tera-Scale Translation Models via Pattern Matching," in *Proceedings of COLING*, Manchester, UK, 2008.

[29] L. Gong, A. Max, and F. Yvon, "Towards Contextual Adaptation for Any-text Translation," in *Proceedings of IWSLT*, Hong Kong, 2012.

[30] J. Bourdaillet and P. Langlais, "Identifying Infrequent Translations by Aligning Non Parallel Sentences," in *Proceedings of AMTA*, San Diego, USA, 2012.

Incremental Unsupervised Training for University Lecture Recognition

Michael Heck[1]*, Sebastian Stüker*[1]*, Sakriani Sakti*[2]*, Alex Waibel*[1]*, Satoshi Nakamura*[2]

[1]International Center for Advanced Communication Technologies (interACT),
Institute for Anthropomatics, Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]Augmented Human Communication Laboratory,
Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan
{heck|sebastian.stueker|alexander.waibel}@kit.edu, {ssakti|s-nakamura}@is.naist.jp

Abstract

In this paper we describe our work on unsupervised adaptation of the acoustic model of our simultaneous lecture translation system. We trained a speaker independent acoustic model, with which we produce automatic transcriptions of new lectures in order to improve the system for a specific lecturer. We compare our results against a model that was trained in a supervised way on an exact manual transcription.

We examine four different ways of processing the decoder outputs of the automatic transcription with respect to the treatment of pronunciation variants and noise words. We will show that, instead of fixating the latter informations in the transcriptions, it is of advantage to let the Viterbi algorithm during training decide which pronunciations to use and where to insert which noise words. Further, we utilize word level posterior probabilities obtained during decoding by weighting and thresholding the words of a transcription.

Index Terms: lecture translation, spoken language translation, simultaneous translation

1. Introduction

Lectures at universities around the world are often given in the language of the country or region that the respective university is located in. At the *Karlsruhe Institute of Technology* (KIT), for instance, most lectures are held in German. This is often a significant obstacle for students from abroad wishing to study at KIT, as they need to learn German first. In order to be able to truly follow the often complex academic lectures, the level of proficiency in German that the foreign students need to reach is quite high.

While, in principal, simultaneous translations by human interpreters might be a solution to bridge the language barrier in this case, in reality this approach is too expensive. Instead, technology in the form of *spoken language translation* (SLT) systems can provide a solution, making lectures available in many languages at affordable costs. Therefore, one of our current research focuses is the automatic translation of university lectures [1][2], and thus aiding foreign students, by bringing simultaneous speech translation technology into KIT's lecture halls.

The simultaneous lecture translation system that we use is a combination of an *automatic speech recognition* (ASR) and a *statistical machine translation* (SMT) system. For the performance of such a *spoken language translation* (SLT) system the *word error rate* of the ASR system is critical, as it has an approximately linear influence on the overall translation performance [3].

Automatic speech recognition for university lectures is rather challenging. In order to obtain the best possible ASR performance, the recognition system's models, including *acoustic model* (AM) and *language model*, need to be tailored as closely as possible to the lecturer's speech and the topic of the lecture.

In this paper we investigate the unsupervised adaptation of the acoustic model of our simultaneous lecture translation to specific speakers. We start with a speaker independent acoustic model that has only seen very few or no data for the respective lecturer to which we adapt. With this model we produce automatic transcriptions of new lectures from one lecturer which we then exploit in order to improve the system for this lecturer. We further examined the impact of various ways of treating pronunciation variants and noise models during model training, as the decoding results on the training data contain those informations besides the hypothesized string of words. However, we will show that it is not necessarily the best strategy to directly use these informations as provided by the recognizer, and rather let the Viterbi algorithm during training decide where to use which pronunciations and when to insert additional noise words.

Similar to [4] we intended to evaluate the possible improvements of a system by unsupervised acoustic model training in dependency of the amount of training data. We share the same basic conditions, that no closely related texts were available for any kind of supervision. Similar to [5, 6], we made use of state confidence scores on word level. As a pre-processing step to unsupervised training, automatic transcriptions were filtered by using word posterior confidence scores for thresholding. Our training conditions can be compared to [7] where new data for retraining comes from the same speaker, channel and related conversation topics. Following the implications of [8] we add low confidence score

data to the training, but unlike in other work we apply word-based weighting in order to compensate for errors, as it was done by [9] for acoustic model adaptation. The assumption is that erroneous data is helpful to improve system generalization. Unlike other work, e.g. [10], we refrained from a lattice-based approach.

2. Data

The experiments in this paper were conducted with the help of the *KIT Lecture Corpus for Speech Translation* [11]. The corpus consists of recorded scientific lectures that were held at the Karlsruhe Institute of Technology (KIT). Currently the corpus mainly contains computer science lectures, and a small amount of lectures from other departments and ceremonial talks.

2.1. Training Data

The speaker-independent system that we used in our experiments was trained on about 94 hours of speech from the lecture corpus. Our experiments were constrained to two distinct speakers. As training data we had 7.4 hours for *speaker A* and 8.3 hours for *speaker B* respectively, which had not been used for training the speaker independent system (see also Section 3).

2.2. Test Data

For *speaker A* we took one, for *speaker B* two recordings — 0.5h and 0.6h overall length respectively — from the available data as our test material. These recordings come from separate lectures than the remaining training data, so that we can actually simulate the way the training data would be used during the real operation of the lecture translator.

3. Experimental Set-Up

In our experiments we simulate the way an ASR system would work when being used in our simultaneous lecture translation system as it is deployed in KIT's lecture halls.

When the system starts to translate the lecture series of a new lecturer, only a generic, mostly speaker independent acoustic model will be available. With every new lecture given, new audio recordings of the lecturer become available, but no manual transcripts. The system will thus only be able to exploit these audio recordings to incrementally transform the speaker-independent acoustic model that is available at the beginning into a speaker-dependent model that fits the specific lecturer.

3.1. Speaker-Independent System

The speaker independent system used in our experiments was taken from the inauguration of the lecture translation system at KIT on June 11th 2012 [12]. For the inauguration, first a speaker-independent acoustic model system was trained on all available training data from the KIT lectures corpus, and then adapted to the individual lecturers.

The ASR system's pre-processing uses the warped minimum variance distortionless response (MVDR) [13] with a model order of 22 without any filter-bank. Vocal tract length normalization (VTLN) [14] was applied in the warped frequency domain. The mean and variance of the cepstral coefficients were normalized on a per-utterance basis. The resulting 20 cepstral coefficients were combined with seven adjacent frames to a single 300 dimensional feature vector that was reduced to 40 dimensions using linear discriminant analysis (LDA).

The acoustic model is based on HMMs using context dependent generalized quinphones with three states per phoneme, and a left-to-right topology without skip states. It uses a total of 4,000 models that were trained using *incremental splitting of Gaussians* (MAS) training, followed by *semi-tied covariances* training [15] and 2 iterations of Viterbi training.

The 4-gram language model used in our experiments was trained on texts from various sources like webdumps, newspapers and acoustic transcripts. The, in total, 28 text corpora range in size from about 5 MByte to just over 6 GByte [12].

4. Unsupervised Training Experiments

In order to adapt the speaker independent acoustic models to our test speakers, we used unsupervised training. For this, the training data of the respective speaker was automatically transcribed. With the help of word lattices, every word in the transcription is annotated with its posterior probability as a measure of confidence. On the transcriptions obtained this way we then performed one iteration of Viterbi training, starting with the speaker independent acoustic model.

In our experiments, described below, we investigated different ways of treating pronunciation variants and noise models in training, as well as different ways of making use of the confidence annotations.

We were also interested in the way that increasing amounts of available training data affect the word error rate. We therefore divided our training data into five chunks (2.29h, 3.05h, 3.81h, 4.57h, 6.87h) for *speaker A* and six chunks (0.99h, 2.17h, 3.44h, 4.79h, 6.23h, 7.68h) for *speaker B*.

We measured the word error rate of the resulting acoustic models on the test set of our test speaker. We decoded the speaker specific test sets with an offline set-up in a similar way decoding is performed in the lecture translation system, i.e., without lattice rescoring, in real-time, and with incremental VTLN and feature space constrained MLLR [16].

4.1. Baseline

A lower limit for the performance of the speaker dependent models that were trained on the unsupervised data is given by the performance of the speaker independent model. It gives a word error rate of 19.7% on our test *speaker A*, and 34.8%

on *speaker B*.

For *speaker A* we were able to estimate how effective the unsupervised training is by comparing our results against a model that was trained in a supervised way on an exact manual transcription of the training data. We expect that this will give us an upper limit for the results obtained from unsupervised training. Just like it was done for the systems for the lecture translation inauguration (see Section 3) we applied one Viterbi training iteration for our test speaker, resulting in a speaker dependent model. It gives a word error rate of 17.3% on the test speaker's test set. For *speaker B*, no exact manual transcriptions were available, rendering these respective tests a case study for the real life application of our system.

4.2. Treatment of Pronunciation Variants and Noise Words

In our first set of experiments we examined how to treat pronunciation variants and noise models in training. The intention of these experiments was to elaborate, whether additional information carried by the transcriptions is beneficial for the training process.

While the output of the recognition run on the training data contains pronunciation variants and noise words, we will see that it is not necessarily the best procedure to use them as provided by the recognizer. Instead it turns out that it is of advantage to let the Viterbi algorithm during training decide which pronunciations to use for the words during training, as well as, where to insert which noise words. This is done by inserting pronunciation variants as alternative paths to the base form of the words; noise words are inserted as alternative paths between regular words.

JANUS allows modifications on the generated hypothesis during label writing. Within the process of writing labels, the decoder chooses the most probable variant of a recognized word and autonomously inserts optional words into the current hypothesis. We experimented with four different ways of processing the decoder outputs of the automatic transcription with respect to the treatment of pronunciation variants and noise words:

recognition The annotation of noise words and pronunciation variants are taken as is from the recognition output and is not altered by the Viterbi training.

baseAll Pronunciation variants in the recognizer output are mapped to their base form, and the pronunciation variants used during training are picked by the Viterbi alignment in training. Wherever a noised word was hypothesized, all other noised words are inserted as alternative paths, and the actual noise word used for training is again picked by the Viterbi alignment.

baseWords Only regular words are mapped to their baseform and their pronunciation variants are inserted as alternative paths. The hypothesized noise words are left as recognized.

filtered All regular words are mapped to their base form, their pronunciation variants are inserted as alternative paths; all recognized noise words are removed, and instead inserted as alternative paths between regular words.

Filtering	Example
filtered	*wenn wir hier*
baseAll	*$ wenn wir $ hier*
baseWords	*$(noise) wenn wir $(breath) hier*
recognition	*$(noise) wenn(1) wir(6) $(breath) hier*

Table 1: Different filtering methods for pre-processing. *filtered* corresponds to plain text, *baseAll* contains general noise tags, *baseWords* is enhanced by annotations of pronunciation variants, and *recognition* resembles the unprocessed decoder output.

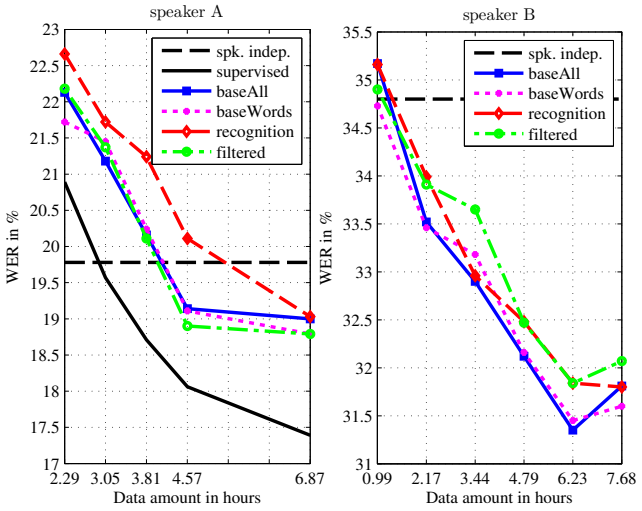


Figure 1: WER in % for the four different configurations for treating pronunciation variants and noise words with increasing amounts of training data.

Table 1 illustrates the different filtering methods that are applied to the automatic transcriptions before training. Figure 1 shows the resulting performance of the four configurations on the test speakers for increasing amounts of training data. The observations allow us to conclude that training on the exact transcriptions from the recognition run (*recognition*) are not the optimal choice. For speaker *A*, the improvement is significantly slower with increasing amounts of training data than for the other three methods. For speaker *B*, this type of transcriptions again does not lead to the optimal training performance. When using all available training material, differences between the filtering methods decrease. However,

recognition type transcriptions can still not keep up with alternatively pre-processed annotations. Categories *baseWords* and *baseAll* perform about equally well, where the latter might be slightly more robust, as the performance curve as a function of the amount of training data tends to stay more stable in comparison to the one of *baseWords* and the other modalities' curves. It is interesting to see that configuration *filtered* seems to be beneficial for systems that already perform reasonably well, whereas for a weaker baseline performance other configurations are preferrable.

The baseline performance seems to have a noticeable impact on the effect of the adaptive training. For systems that already perform well, the improvements that can be expected tend to be lower than for systems that start with poorer recognition capabilities, according to the observations: For speaker *B*, speaker dependent models perform better than the speaker-independent models when at least 1h of training data is available, whereas for speaker *A* about four times as much data is necessary to see improvements in the same magnitude. The better the baseline performs, the more data is needed to observe first improvements. Ultimately and as expected, training on exact, i.e., manual transcriptions of the training data outperforms the unsupervised training, as can be seen for speaker *A*, where we had manual annotations at hand.

4.3. Confidence Weighting & Thresholding

The most common methods for processing unreliable, erroneous transcriptions in unsupervised acoustic model training are based on lattice confidence measures at word or state level [10]. In our experiments we utilized the word level posterior probabilities obtained during decoding. We utilized the confidences in three ways:

weighted Sets the gamma probabilities of the states of a word during Viterbi training to the posteriori probability of the respective word.

thresh Removes words with a confidence below a certain threshold from training.

weighted+thresh Combines both methods.

Given a Viterbi path through a built up HMM of a training utterance, a weighting factor *gamma* can be assigned to every frame prior to the update step for the model weights. If *gamma* is set to 0, parts of a path are effectively excluded from training. A $gamma \neq 0$ results in a weighted contribution of this particular frame to the training. Here, the *gamma* value corresponds to the confidence score $conf(w)$, e.g., the posterior probability of the word w to which a frame fr_i^w belongs, if weighting is applied. Thresholding with a factor *t* is performed by setting *gamma* to 0 for each fr_i^w with $conf(w) \leq t$.

Figure 2 shows the result of weighting with confidences and applying a threshold for our test speakers. Of these methods, the word-based weighting produces the better systems for both speakers: Weighting with the confidences gives the best performance, particularly when using all availabe data, whereas thresholding at least matches the performance of training on unfiltered data and at best gives slight advantages over not using confidences at all, in cases where a sufficient amount of training data is available. We randomly selected a threshold of 25% as lower limit for our experiments. Further experiments that are not represented in Figure 2 revealed, that the threshold should be used with care, as using too high a threshold discards too much data, and the performance suffers. Combining weighting and a threshold of 25% leads to the smoothest performance curve and has an effect on performance similar to weighting alone.

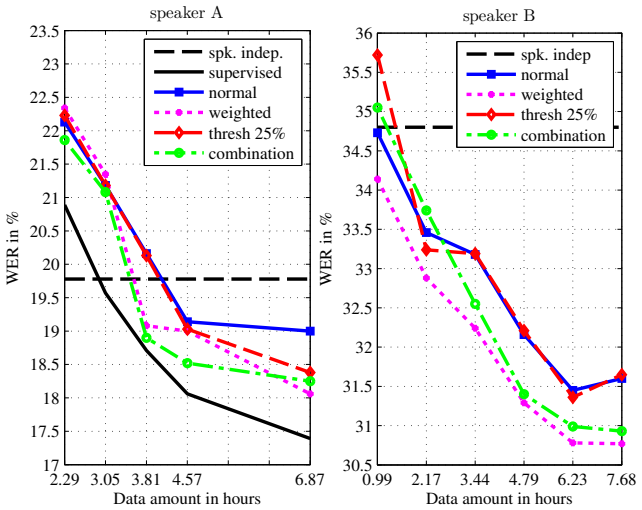


Figure 2: WER in % when weighting training data with confidences (*weighted*), excluding words from training by a posterior probability below a certain threshold (25%) (*thresh 25%*), and a combination of both (*combination*), compared to training without weighting and thresholding (*normal*).

5. Conclusion

In this paper we have described work on unsupervised adaptation of the acoustic model of our simultaneous lecture translation. We produced automatic transcriptions of new lectures with a speaker independently trained baseline system in order to improve the same for specific lecturers.

Evaluating four different ways of processing the decoder outputs led to the conclusion that it is of advantage to let the Viterbi algorithm during training decide which pronunciations to use and where to insert which noise words, instead of fixating the latter informations. The degree of detail for the transcriptions correlates with the baseline performance given a target speaker. *baseAll* and *baseWords* are beneficial when the baseline performance is lower, whereas *filtered* is better when the system already performs reasonably well, promoting the expectation that the Viterbi algorithm is able to make more accurate decisions with a better starting point,

whereas additional information provided with the transcriptions helps when the baseline models show a lower performance. Speaker dependent models perform better than the speaker-independent models when at least 1h of training data is used. The Viterbi algorithm needs a certain amount of data so that training will succeed, where the amount required for performance gains correlates with the baseline performance.

Further, we utilized the word level posterior probabilities obtained during decoding by weighting and thresholding the words of a transcription. Combining word-based weighting and a threshold of 25% led to the smoothest performance curve as a function of the amount of training data. Our best systems in terms of WER reach an error rate of 18% and 30.7% for speakers *A* and *B* respectively, being trained on *baseAll* (*A*) and *baseWords* (*B*) processed transcriptions and *weighted* training. An additional *threshold* of 25% led to a competitive WER of 18.2% and 30.9% respectively. Obviously, weighting allows to cushion the influence of erroneously annotated training data, which is likely to have a lower confidence than potentially correct parts. The same explanation applies for the combination of weighting and thresholding, which leads to an even smoother convergence, whereas thresholding alone either excludes only few erroneous data, or even lots of correct data, depending on it's strictness. The winning techniques represent possible candidates for use in our simultaneous lecture translation systems as they combine fast convergence with good performance.

6. Acknowledgements

This work was supported in part by an interACT student exchange scholarship. '*Research Group 3-01*' received financial support by the '*Concept for the Future*' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative. The work leading to these results has received funding from the European Union under grant agreement no. 287658.

7. References

[1] C. Fügen, A. Waibel, and M. Kolss, "Simultaneous translation of lectures and speeches," *Machine Translation*, vol. 21, pp. 209–252, 2007.

[2] C. Fügen, "A system for simultaneous translation of lectures and speeches," Ph.D. dissertation, Universiät Karlsruhe (TH), November 2008.

[3] S. Stüker, M. Paulik, M. Kolss, C. Fügen, and A. Waibel, "Speech translation enhanced asr for european parliament speeches - the influence of asr performance on speech translation," in *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4. Honolulu, HI, USA: IEEE, April 2007, pp. 1293–1296.

[4] L. Lamel, J. luc Gauvain, and G. Adda, "Unsupervised acoustic model training," in *Proceedings of the ICASSP 2002*, Orlando, Florida, USA, May 2002.

[5] C. Gollan, S. Hahn, R. Schlüter, and H. Ney, "An improved method for unsupervised training of lvcsr systems," in *Proceedings of the INTERSPEECH 2007*, Antwerp, Belgium, August 2007.

[6] T. Kemp and A. Waibel, "Unsupervised training of a speech recognizer using tv broadcasts," in *Proceedings of the EUROSPEECH 1999*, Budapest, Hungary, September 1999.

[7] G. Zavaliagkos, M.-H. S. abd Thomas Colthurst, and J. Billa, "Unsupervised acoustic model training," in *Proceedings of the ICSLP 1998*, Sydney, Australia, November 1998.

[8] H. Li, T. Zhang, and L. Ma, "Confirmation based self-learning algorithm in lvcsr's semi-supervised incremental learning," *Procedia Engineering*, vol. 29, pp. 754–759, 2012.

[9] C. Gollan and M. Bacchiani, "Unsupervised acoustic model training," in *Proceedings of the ICASSP 2008*, Las Vegas, NV, USA, March 2008.

[10] T. Fraga-Silva, J.-L. Gauvain, and L. Lamel, "Lattice-based unsupervised acoustic model training," in *Proceedings of the ICASSP 2011*, Prague, Czech Republic, May 2011.

[11] S. Stüker, F. Kraft, C. Mohr, T. Herrmann, E. Cho, and A. Waibel, "The kit lecture corpus for speech translation," in *LREC 2012*, Istanbul, Turkey, May 2012.

[12] E. Cho, C. Fügen, T. Hermann, K. Kilgour, M. Mediani, C. Mohr, J. Niehues, K. Rottmann, C. Saam, S. Stüker, and A. Waibel, "A real-world system for simultaneous translation of german lectures," interACT, Karlsruhe Institute of Technology, Tech. Rep., December 2012.

[13] M. Wölfel and J. McDonough, "Minimum variance distortionless response spectral estimation, review and refinements," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 117–126, September 2005.

[14] P. Zhan and M. Westphal, "Speaker normalization based on frequency warping," in *ICASSP*, Munich, Germany, April 1997.

[15] M. Gales, "Semi-tied covariance matrices for hidden markov models," Cambridge University, Engineering Department, Tech. Rep., February 1998.

[16] M. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," Cambridge University, Engineering Department, Tech. Rep., May 1997.

Studies on Training Text Selection for Conversational Finnish Language Modeling

Seppo Enarvi and Mikko Kurimo

Aalto University
School of Electrical Engineering
Department of Signal Processing and Acoustics
seppo.enarvi@aalto.fi

Abstract

Current ASR and MT systems do not operate on conversational Finnish, because training data for colloquial Finnish has not been available. Although speech recognition performance on literary Finnish is already quite good, those systems have very poor baseline performance in conversational speech. Text data for relevant vocabulary and language models can be collected from the Internet, but web data is very noisy and most of it is not helpful for learning good models. Finnish language is highly agglutinative, and written phonetically. Even phonetic reductions and sandhi are often written down in informal discussions. This increases vocabulary size dramatically and causes word-based selection methods to fail. Our selection method explicitly optimizes the perplexity of a subword language model on the development data, and requires only very limited amount of speech transcripts as development data. The language models have been evaluated for speech recognition using a new data set consisting of generic colloquial Finnish.

1. Introduction

Finnish language has a colloquial variant that differs from the formal literary Finnish substantially. While clearly pronounced literary Finnish can already be recognized with high precision, current ASR systems are unable to recognize conversational Finnish, because there has not been any training or evaluation data available.

With regard to a speech recognizer, the set of phonemes is the same in both language varieties, but the difference in vocabulary and grammar is clear [1], so we have started the research on colloquial Finnish NLP by collecting text data. The relevance of the collected text for speech recognition has been evaluated with Aalto speech recognizer. In addition to speech recognition, the data is valuable for other tasks such as machine translation as well, because Finnish language communication more and more includes colloquial characteristics [2].

So far there are no statistical language models that would cover colloquial Finnish. Finnish conversations in e.g. Internet are written down phonetically, often including phoneme reductions and compounding, suggesting that on-line discussions would offer useful data for language modeling.[1] While there are huge amounts of data available, it is important to select only what is useful for the modeling task. The irrelevant n-grams increase confusability and computational burden in language models. Irrelevant data also makes analysis such as discovery of morphemes and word classes error-prone and computationally more intense. For these reasons we have evaluated speech recognition errors and language model perplexities, as well as the reduction in data size.

Related research has been carried out earlier in the context of adapting an out-of-domain language model with in-domain data. A popular approach has been to train an in-domain language model and select text segments with low perplexity [3]. Klakow trained language models from out-of-domain data, computing the change in in-domain perplexity, when a text segment is removed from the training data [4].

Sethy et al. used relative entropy to match the distribution of the filtered data with the in-domain distribution [5]. Instead of scoring and filtering each text segment individually, they select text segments sequentally, adding a new segment to the selection if it reduces relative entropy with respect to the in-domain data. The algorithm was later revised to use a smoothed version of the Kullback-Leibler distance that uses a tunable smoothing parameter [6], with improved results.

Moore and Lewis used formal reasoning to show that if the selection method is based on the probability (in terms of cross-entropy or perplexity) given by an in-domain language model to the training text segment, one should compare the probability to the probability given by an out-of-domain language model [7]. They computed the cross-entropy of each text segment according to an in-domain language model and an out-of-domain language model, and used the difference between the two cross-entropies as the selection criterion.

From the above approaches the one proposed by Klakow requires the least amount of in-domain development data, since models are estimated only from the out-of-domain data. At the time we had very little in-domain development data of conversational Finnish (we used a set of 1047 utterances in these experiments), so this was the only applicable approach. The method may become computationally demanding since it

[1] The most notable difference between transcribed speech and written conversation is that disfluencies are usually omitted in writing.

requires training as many language models as there are text segments, but the computation can be done in parallel.

Another line of research has used information retrieval techniques to select in-domain documents. Term frequency–inverse document frequency (tf-idf) is a popular measure of document similarity. After constructing a vector representation of each document, it is efficient to find documents that are similar to a query string. Mahajan et al. proposed to use it for language model adaptation based on current recognition history [8].

We have collected Internet conversations using Google search, and by crawling Finnish discussion sites. The obtained text segments are scored, and the worst scoring segments are pruned. The threshold score for pruning text segments is found automatically, so as to minimize the perplexity of the resulting language model on a held-out data set. The unlimited vocabulary presents challenges in using perplexity for scoring and for finding the pruning threshold. The perplexity optimization is possible only with a subword language model. We have also collected a new set of transcribed Finnish conversations for development and evaluation purposes.

The next section discusses the challenges posed by the unlimited nature of Finnish vocabulary. Section 3 presents our new development and evaluation data. In Section 4 we explain how we have collected web data for language modeling. Section 5 describes how we have performed our evaluations, and the results are given in Section 6. Finally, Section 7 draws conclusions.

2. Vocabulary in conversational Finnish speech recognition and perplexity computation

The highly agglutinative nature of Finnish language makes it difficult to create an exhaustive vocabulary for speech recognition. Creutz et al. show a comparison of vocabulary growth across different languages [9]. While conversational speech is generally thought to be less diverse than planned speech, there are no less word forms used in conversational Finnish text than in a similar amount of literary Finnish. The reason is that the phonetic variation in conversational Finnish is translated into new vocabulary.

Finnish orthography is very close to phonemic, meaning that written letters generally correspond to spoken phonemes. In informal conversations, phonetic variation is also often reflected in writing, even to the extent that sandhi is expressed in written form. For example, “en minä tiedä” is literary Finnish, and can be translated “i don’t know”. Reduced forms of the same expression are used in spoken conversation, but often in textual communication as well:

en mä tiedä
en mä tiiä
emmä tiiä

The situation is different from English, where there is generally only one way to spell each word, even though several different pronunciations exist. When having a spoken conversation, one could actually utter /aɪ doʊnt noʊ/, /ədənoʊ/, or /dʌnəʊ/, but in any of these cases one would probably write “i don’t know”, if the conversation was textual.

A comparison of vocabulary growth in Finnish and English Internet conversations and formal texts is shown in Figure 1. The formal English plot has been created from newspaper corpora. The formal Finnish data is literary Finnish from books and newspapers. The conversational Finnish text is Internet conversations from Suomi24 discussion site, covering many different topics. The conversational English is gathered from the web by searching text related to topics in meeting transcripts from CMU, ICSI, and NIST [10]. Web texts were processed using normalization scripts. It should be noted that the quality of text normalization may vary, as well as the degree to which the web data sets are spontaneous and colloquial.

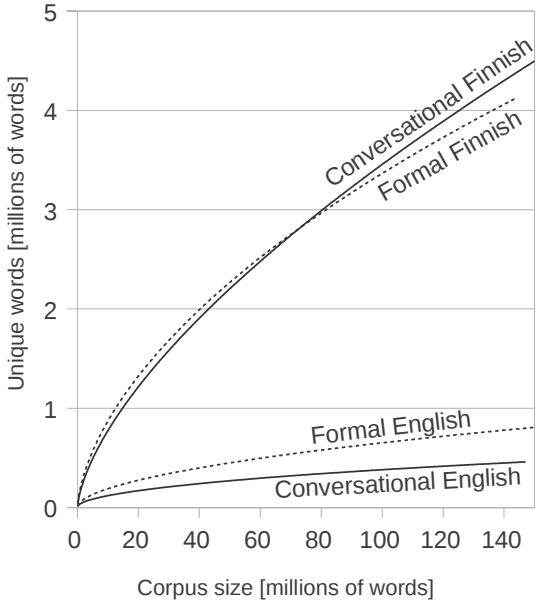


Figure 1: Vocabulary growth, when all the encountered words are added to the vocabulary, on newspaper-style formal text and Internet conversations

The curves show that, as expected, vocabulary growth in formal English is clearly faster than that in English conversations. However, in Finnish Internet conversations vocabulary grows at a similar pace to, and eventually exceeds that of formal Finnish.

Another comparison was made to see how the vocabulary growth affects OOV rates, by using an independent test set from each category. Figure 2 illustrates the percentage of words in the corresponding test set that are missing from the training set, for growing amounts of training data. The training data is the data used to plot Figure 1. The formal English test data was transcribed broadcast news speech, and the formal Finnish was planned literary Finnish from the SPEECON [11] corpus. The conversational test data sets

were transcribed conversations, omitting hesitations. The high OOV rates on transcribed Finnish conversations suggest that the vocabulary growth in Finnish Internet conversations is not just a result of poor text normalization or clean up.

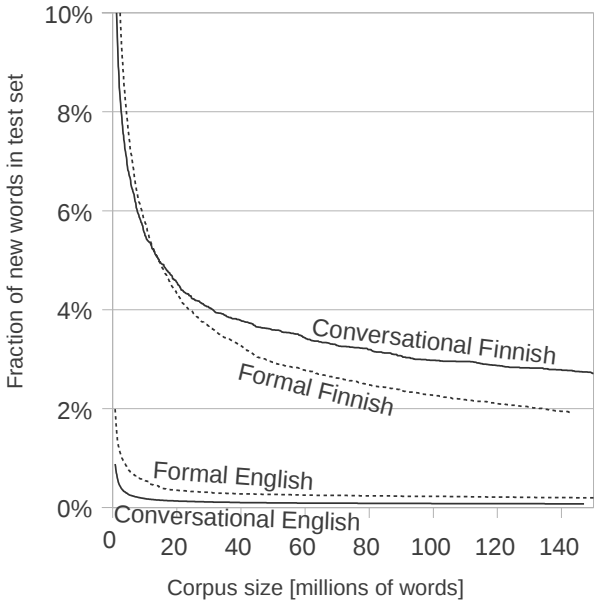


Figure 2: Development of OOV rate, when all the encountered words are added to the vocabulary, on newspaper-style formal text and Internet conversations

The standard approach for unlimited vocabulary Finnish language speech recognition has been to use statistical morphs as the basic language modeling unit, instead of words [12]. It seems that statistical morphs obtained by direct application of Morfessor Baseline [13] to the word list do not model conversational Finnish well. The reason may be insufficient quality or quantity of training data, or the pronunciation variation behind new word forms. Factored language models [14] is one way to alleviate the vocabulary size issue, but at the moment there are no tools for extracting meaningful factors from colloquial Finnish word forms. Development of such tools would be extremely difficult because of the numerous ways in which phonetic variation can alter the words.

We tried conversational speech recognition with morph-based models, but so far there was no improvement over word models in terms of word error rate. However, the perplexity computations in the text selection algorithm have been performed using morph models. The reason is that there are so many OOV words that we need reliable estimates also for n-grams containing OOV words. Even though the initial morph models are not yet sufficiently good for ASR, they seem to offer a reasonable approximation for perplexity computation.

Language model perplexity is generally computed either including only those words that occurred in the training data, or using an open-vocabulary language model, i.e. one that contains the unknown word token <UNK>. The probability for unknown words is obtained by replacing the most infrequent words in the training data with <UNK>, or by discounting the observed unigram probabilities.

If one chooses to use a closed vocabulary, and compute perplexity only on in-vocabulary n-grams, the perplexity value will increase when the number of OOV words decreases. This makes perplexity optimization in Finnish difficult, since we do not know if we should prefer low perplexity or low OOV rate. This problem is easily overlooked with English language data, because the percentage of OOV words stays constant enough not to play a significant role in determining the perplexity value.

We did not find open-vocabulary language models to be a suitable solution either. The problem is that the selection algorithm is significantly affected by how the <UNK> probability is determined. The collected conversational Finnish text contains so many word forms that occur only once, that their probability mass alone gives a too high estimate for the OOV probability. Selection of text segments based on perplexity of such a model would prefer segments with high OOV rate.

3. Transcribed Finnish conversations

For development and evaluation data, we have transcribed Finnish conversations: five radio conversations from 13 different speakers, three podcast conversations from 5 different speakers, and recordings of 67 students discussing in pairs with headsets on. These conversations encompass a diverse set of speaking styles and topics, as the intention of this research is not to adapt statistical models to a specific topic or domain, but to collect generic colloquial Finnish data. The students were encouraged to discuss from any topic they could think of, although they were given 16 example topics. They could also use a web browser to find conversation topics from news sites. Only a portion of each conversation containing fluent conversation was selected for transcription. The discussions were entirely colloquial and very natural.

The conversations were divided into development and evaluation sets so that the same radio programs or speakers do not appear in both development and evaluation set. In total the evaluation set contains 44 minutes of audio, 541 utterances, and 17 different speakers. DEVEL1 development set contains 1047 utterances from 49 speakers, and DEVEL2 contains 445 utterances from 19 speakers.

Development data is required for filtering out irrelevant text. It may be used for both scoring text segments, and optimizing the rejection threshold, as explained in the next section. It is essential that in such case, different development data is used for text scoring and for finding the threshold score. Otherwise filtering will be too intense because of overfitting. When development data is not needed for text scoring, we have used the entire DEVEL1 and DEVEL2 data sets for optimizing the rejection threshold. Otherwise DEVEL1 has been used for scoring, and DEVEL2 has been held out for optimizing the filtering threshold. Both DEVEL1 and DEVEL2 sets were also included as training data for the acoustic model used in the speech recognition experiments.

4. Collecting and filtering web text for modeling Finnish language

4.1. Collected web corpora

Internet search engines are commonly used to query text for language modeling [10]. We collected several text corpora from the Internet, first using a script that extracts results from Google queries. The data set WEB1 was retrieved using devised 2-grams, 3-grams and 4-grams as query strings. The query n-grams were constructed from colloquial word forms, intentionally forming expressions that are used only in conversational Finnish.

A more systematic way to gather data set WEB2 was used. We extracted all the 3-grams from a transcribed radio conversation, and those that exist in a literary Finnish corpus were removed. The remaining 667 3-grams were used as search queries. We did not try other n-gram lengths, but 4-grams rarely return more than a few search results, and 2-grams are often too generic, returning even other than Finnish text. Surprisingly, WEB2 data did not improve recognition performance. Also, without a substantial amount of existing in-domain text, the amount of data obtained with this method was still small.

Data set WEB3 was extracted by copying the entire contents of a web site containing Internet Relay Chat (IRC) conversations. Data sets WEB4 and WEB5 were each collected by crawling a Finnish discussion site using Python libraries Scrapy and Selenium, and extracting every conversation. This turned out to be a fast method for obtaining large amounts of structured data.

4.2. Preprocessing web text

Extensive preprocessing was needed, before the web data could be used for language modeling. This included

- removal of non-textual items, such as hyperlinks, message board markup code, usernames, and smileys,
- expansion of abbreviations, numbers, punctuation marks, and such, and
- deletion of words that contain phoneme sequences that do not pertain to Finnish phonological rules.

Numbers do not carry information about pronunciation. We have simply expanded them as they are pronounced in literary Finnish. The sizes of the data sets after preprocessing are shown in Table 1.

Data set	Number of words
WEB1	767,669
WEB2	1,067,993
WEB3	562,426
WEB4	25,131,015
WEB5	46,258,268
DEVEL1	17,209
DEVEL2	8,755

Table 1: Data sets and their sizes after preprocessing

4.3. Text segment scoring

Data filtering starts by giving a numeric score to each text segment. Then segments whose score is below a threshold will be rejected from the training data. A shortcoming of this one-pass scheme is that every example of a common, short sentence receives the same high score, which may skew the distribution of the selected data too much towards frequent utterances such as "okay" [5, 6]. For this reason, the segments that we score are web pages and discussion site messages, rather than sentences.

Among colloquial Finnish, the collected corpora contained literary Finnish, foreign language, and even garbage such as HTML code that had slipped through the preprocessing scripts. The following scoring methods were targeted to separate such noise from the relevant text segments.

- **avg-unigram-count**. Word unigram counts are calculated from the entire text data. The score of a text segment is the average of the counts of the words in the segment.
- **median-unigram-count**. The score of a text segment is the median of the counts of the words in the segment. The reasoning is that garbage segments often contain short words that by chance are very common in Finnish language, and increase average unigram count.
- **devel-lp-ngram**. An n-gram model is estimated from the entire training data, and with a segment removed. The decrease in development data log probability when a segment is removed, is the score of the segment. This is the selection criterion used by Klakow [4].
- **devel-lp-ngram-topic**. As *devel-lp-ngram*, but filtering is applied per discussion site conversation instead of per discussion site message. Longer text segments allow more reliable probability estimates, but less fine-grained filtering.

4.4. Finding optimal filtering threshold

After every text segment has been assigned a score, those that have a score below a filtering threshold, will be excluded from the training data. We optimize a different threshold for every corpus, using the following method: The segments are sorted from the highest scoring to the lowest scoring. The training set is grown by gradually including more and more text, starting from the highest scoring segment. A bigram morph model is estimated from the training set, and development set perplexity is computed, at frequent intervals. Then we find the threshold score that minimizes perplexity.

It would be computationally too expensive to resegment the vocabulary into morphs every time training data is increased. We found it adequate to segment each corpus once, even though this means that with less training data, not all the morphs necessarily occur in the language model. The number of OOV morphs is significant only with very little training

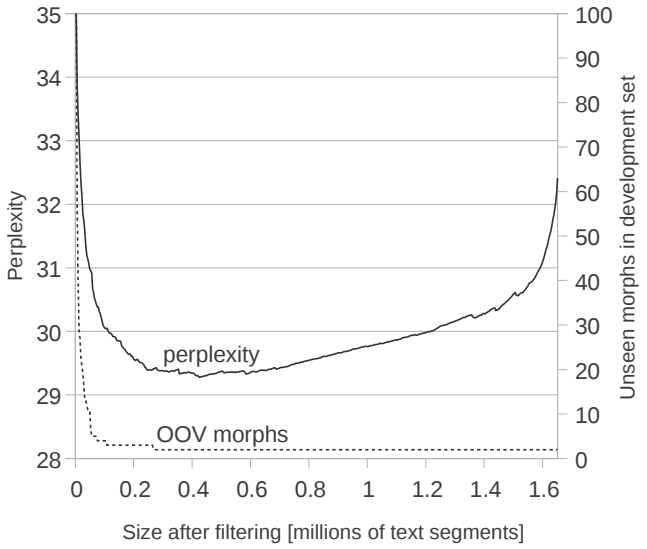


Figure 3: Perplexity and the number of OOV morphs on a held-out data set, with a growing amount of training data included in the order of *devel-lp-1gram* score

data. Figure 3 shows how perplexity and OOV rate behave as a function of included training data, with a fixed morph segmentation.

5. Experimental setup for language model evaluation

5.1. Speech recognizer

The speech recognition experiments were carried out using Aalto ASR system [15]. Our baseline model for recognizing standard Finnish has been trained on planned speech from the SPEECON [11] corpus. The model used in these experiments was trained on the SPEECON data, augmented with 176 minutes of our new development data, and 622 minutes of audio from FinDialogue, the conversational part of the FinINTAS corpus [16].

5.2. Error measure

Phonetic variation also creates challenges when measuring recognition accuracy. As most of the words can be pronounced in several slightly different ways, and the words are written out as they are pronounced, it would be harsh to compare recognition against the verbatim phonetic transcription. Thus word forms that are simply phonetic variation were added as alternatives in the reference transcriptions. This caused a large amount of manual work in top of transcription, since the added alternative pronunciations depend also on the meaning of the word, i.e. the context needs to be considered when adding alternations.

It has been customary in Finnish language speech recognitions to use letter error rate (LER) as the measure of speech recognition accuracy. We are not yet sure how to implement LER in the presence of a large number of alternative hypotheses of varying length, so this paper uses word error rate (WER).

5.3. Language models

Simply concatenating the data sets to estimate a language model would result in a model that is dominated by the biggest corpora, and performs poorly. A popular approach to combining different corpora is by linear interpolation of the language model probabilities. With many corpora, this becomes inefficient, requiring the decoder to evaluate every model for each possible word expansion. We used an approximative approach, where the probabilities of all observed n-grams are obtained by interpolating component model probabilities, and the remaining probabilities are computed to normalize the model [17]. The component weights are computed by optimizing development data (DEVEL1 + DEVEL2) perplexity. All the language models used in these experiments were pruned by removing n-grams whose removal caused less than 5×10^{-10} increase in training data perplexity.

We wanted to eliminate the effect of vocabulary selection from the data selection experiments, so all the word models were trained with the same 87,971 word vocabulary consisting of the words that occur at least 40 times in data sets WEB1 to WEB5. This left 8.4 % of word tokens in the verbatim evaluation set transcriptions out of vocabulary. However, since the reference transcriptions include alternative word forms, the recognizer may occasionally recognize a word correctly, even if the exact word form is not included in the vocabulary. Taking the alternatives into account, 6.0 % of the evaluation set word tokens could not be recognized with this vocabulary.

6. Results

6.1. Filtering evaluation

Table 2 shows the total size of data sets WEB1 to WEB5 before and after filtering, and error rates given by 4-gram language models on the evaluation data. *devel-lp-1gram* was the most effective filtering method. It resulted in a small data set (23 % of the original word tokens), and 1.0 % reduction in WER. In line with Klakow's results [4] filtering worked slightly better with unigram than bigram log probability.

avg-unigram-count filtering did not improve error rates, on contrary to *median-unigram-count*. Performing filtering only on conversations was too coarse-grained. *devel-lp-1gram-topic* reduced the amount of text and recognition errors minimally.

6.2. Comparison against existing corpora

Our current baseline language models have been created using 143 million words from the Finnish Language Text Collection (FTC), an electronic collection of Finnish text from newspa-

Filtering algorithm	WEB1	WEB2	WEB3	WEB4	WEB5	Interp.	Words
unfiltered	63.6	66.4	65.9	60.4	60.6	59.2	73,787,371
avg-unigram-count	63.5	66.5	65.8	59.8	60.9	59.6	35,426,285
median-unigram-count	63.5	66.1	65.9	59.6	59.9	58.6	37,637,867
devel-lp-1gram	63.4	**65.2**	65.5	59.5	**58.5**	**57.5**	**16,936,104**
devel-lp-2gram	63.5	65.7	**65.3**	**58.9**	59.1	57.7	19,059,831
devel-lp-1gram-topic	**63.3**	66.0	65.6	60.4	60.3	59.1	69,710,151
devel-lp-2gram-topic	63.5	65.7	**65.3**	60.4	60.3	59.5	69,708,077

Table 2: Recognition results from language models trained on filtered and unfiltered web data sets and an interpolated language model, and remaining total training data sizes in words

pers, journals, and books from the 1990's. Word error rates around 10 % on literary Finnish can be achieved with language models estimated from this corpus alone. Recently we have acquired two new corpora: 442,000 word "Helsingin puhekielen korpus" (HPK), a collection of interviews in dialectal language from the 1970's [18], and FinDialogue (FD), 81,000 words of conversational Finnish from FinINTAS corpus [16]. We have evaluated the web data in a speech recognition experiment against these corpora. All these corpora are either available or becoming available from CSC—IT Center for Science in Finland.

The comparison in Table 3 shows how poorly the existing corpora match colloquial Finnish speech. The collected web data alone performs better than the previous corpora combined with interpolation. WEBfilt is the web data after *devel-lp-1gram* filtering. It outperforms the previous corpora by 3.8 % in terms of word error rate. When the web data is combined with the previous corpora, WER is reduced by 7.0 %. This is a clear improvement in performance, given the amount of evaluation data, 44 minutes of speech from 17 speakers.

While filtering improved WER significantly when using only web data, when interpolating with the other corpora, it reduced model size, but did not improve WER. This result suggests that the interpolation may not be optimal. It might be beneficial to filter also the literary Finnish corpora, or try different adaptation techniques.

Training set	N-grams	WER	PPL
FTC	20,780,423	72.2	6364
FTC+HPK+FD	8,772,995	59.8	674
WEB	15,803,759	59.2	652
WEBfilt	3,694,060	57.5	589
FTC+HPK+FD+WEB	14,884,046	**55.6**	**493**
FTC+HPK+FD+WEBfilt	5,429,240	55.7	496

Table 3: Language model sizes, recognition results, and perplexities from models interpolated from existing corpora and the collected web data

By combining the web data with existing corpora, we obtained 55.6 % WER. This can be compared to 61.9 % WER we obtained with acoustic model trained only on SPEECON corpus, using the same language model. The improvement is significant, although we still have only little colloquial Finnish speech data for acoustic model training.

Perplexity can be used to evaluate how well a language model alone performs on colloquial Finnish text. The perplexities in Table 3 were computed on the verbatim transcripts of the evaluation data, i.e. considering only the exact word forms as they were pronounced. They show even greater improvement than the speech recognition experiments, in how well the data sets match the evaluation data, indicating that the poor recognition results may partly be due to the acoustic model trained on mostly literary Finnish matching poorly with conversational speech.

For comparison, we have included some results from morph models in Table 4, although so far we have not been able to get morph-based recognition on par with word-based recognition on colloquial Finnish. The morph results are from interpolated 5-gram morph models. Morph segmentations were computed using Morfessor Baseline (MDL) algorithm [13] from words that occur at least three times in the training data, with equal weight on each word. Resulting morph vocabularies ranged from 107,000 to 130,000 morphs.

Training set	N-grams	WER
FTC+HPK+FD	11,374,836	63.9
FTC+HPK+FD+WEB	10,558,474	**58.8**
FTC+HPK+FD+WEBfilt	5,385,316	59.4

Table 4: Language model sizes and recognition results from morph-based models interpolated from existing corpora and the collected web data

There was no clear difference in the recognition results between the radio conversations and the student conversations. The podcast conversations gave highest error rates, presumably because they contain some uncommon technological jargon.

7. Conclusions

We have collected large amounts of language model training material for colloquial Finnish from the Internet, and pruned it effectively, reducing the data size, language model perplexity, and speech recognition error rates, with very limited development data available. We have also described why the unlimited nature of the vocabulary and pronunciation variation

makes this task, as well as speech recognition, particularly difficult on colloquial Finnish. The standard approach to unlimited vocabulary in Finnish is language models based on subword units. We have found morph-based language models useful in filtering text of a highly agglutinative language, but so far traditional word-based language models have worked best for the recognition task, at least in terms of word error rate. We hope the new cleaned-up data sets will help us collect even more data and address modeling the lexicon and the pronunciation variation of colloquial Finnish in our future research to develop effective statistical models for ASR and MT.

8. Acknowledgement

This work was financially supported by the Academy of Finland under the grant number 251170 (Finnish Centre of Excellence Program (2012–2017)) and Mobster project funded by Tekes.

9. References

[1] Seppo Enarvi, *Finnish Language Speech Recognition for Dental Health Care*, Licentiate thesis, Aalto University School of Science, Espoo, Finland, Mar. 2012.

[2] L. M. Määttänen, "Puheenomaisten piirteiden ilmeneminen erityyppisissä suomalaisissa kirjoitetuissa teksteissä," M.S. thesis, University of Groningen, Groningen, Netherlands, Aug. 2007.

[3] Pablo Fetter, Alfred Kaltenmeier, Thomas Kuhn, and Peter Regel-Brietzmann, "Improved modeling of oov words in spontaneous speech," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1996)*, Washington, DC, USA, 1996, vol. 1, pp. 534–537, IEEE Computer Society.

[4] Dietrich Klakow, "Selecting articles from the language model training corpus," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2000)*. 2000, vol. 3, pp. 1695–1698, IEEE Computer Society.

[5] Abhinav Sethy, Panayiotis G. Georgiou, and Shrikanth Narayanan, "Text data acquisition for domain-specific language models," in *Proc. 2006 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA, 2006, EMNLP '06, pp. 382–389, Association for Computational Linguistics.

[6] Abhinav Sethy, Panayiotis G. Georgiou, Bhuvana Ramabhadran, and Shrikanth S. Narayanan, "An iterative relative entropy minimization-based data selection approach for n-gram model adaptation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 13–23, 2009.

[7] Robert C. Moore and William Lewis, "Intelligent selection of language model training data," in *Proc. ACL 2010 Conference Short Papers*, Stroudsburg, PA, USA, 2010, ACLShort '10, pp. 220–224, Association for Computational Linguistics.

[8] Milind Mahajan, Doug Beeferman, and X.D. Huang, "Improved topic-dependent language modeling using information retrieval techniques," in *Proc. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1999)*. 1999, vol. 1, pp. 541–544, IEEE Computer Society.

[9] Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke, "Morph-based speech recognition and modeling of out-of-vocabulary words across languages," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 5, no. 1, pp. 3:1–3:29, Dec. 2007.

[10] Ivan Bulyko, Mari Ostendorf, Manhung Siu, Tim Ng, Andreas Stolcke, and Özgür Çetin, "Web resources for language modeling in conversational speech recognition," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 5, no. 1, pp. 1:1–1:25, Dec. 2007.

[11] Dorota J. Iskra, Beate Grosskopf, Krzysztof Marasek, Henk van den Heuvel, Frank Diehl, and Andreas Kießling, "SPEECON - speech databases for consumer devices: Database specification and validation," in *Proc. Third International Conference on Language Resources and Evaluation (LREC 2002)*, Canary Islands, Spain, May 2002.

[12] Teemu Hirsimäki, Mathias Creutz, Vesa Siivola, Mikko Kurimo, Sami Virpioja, and Janne Pylkkönen, "Unlimited vocabulary speech recognition with morph language models applied to Finnish," *Computer Speech & Language*, vol. 20, no. 4, pp. 515–541, Oct. 2006.

[13] Mathias Creutz and Krista Lagus, "Unsupervised discovery of morphemes," in *Proc. ACL 2002 workshop on morphological and phonological learning*, Stroudsburg, PA, USA, 2002, vol. 6 of *MPL '02*, pp. 21–30, Association for Computational Linguistics.

[14] Jeff A. Bilmes and Katrin Kirchhoff, "Factored language models and generalized parallel backoff," in *Proc. 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT/NAACL-2003)*, Edmonton, Alberta, May/June 2003, vol. 2 of *NAACL 2003–short papers*, pp. 4–6.

[15] Teemu Hirsimäki, Janne Pylkkönen, and Mikko Kurimo, "Importance of high-order n-gram models in

morph-based speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 724–732, 2009.

[16] Mietta Lennes, "Segmental features in spontaneous and read-aloud Finnish," in *Phonetics of Russian and Finnish. General Introduction. Spontaneous and Read-Aloud Speech*, Viola de Silva and Riikka Ullakonoja, Eds., pp. 145–166. Peter Lang GmbH, 2009.

[17] Andreas Stolcke, "SRILM—an extensible language modeling toolkit," in *Proc. 7th International Conference on Spoken Language Processing (ICSLP 2002)*, 2002, pp. 901–904.

[18] Heikki Paunonen, *Suomen kieli Helsingissä: huomioita Helsingin puhekielen historiallisesta taustasta ja nyky-variaatiosta*, Helsinki: Helsingin yliopiston suomen kielen laitos, 1995.

Assessing Quick Update Methods of Statistical Translation Models

Shachar Mirkin[1], *Nicola Cancedda*[2]

[1] Laboratoire d'Informatique de Grenoble, [2] Xerox Research Centre Europe

shachar.mirkin@imag.fr, nicola.cancedda@xrce.xerox.com

Abstract

The ability to quickly incorporate incoming training data into a running translation system is critical in a number of applications. Mechanisms based on incremental model update and the online EM algorithm hold the promise of achieving this objective in a principled way. Still, efficient tools for incremental training are yet to be available. In this paper we experiment with simple alternative solutions for interim model updates, within the popular Moses system. Short of updating the model in real time, such updates can execute in short timeframes even when operating on large models, and achieve a performance level close to, and in some cases exceeding, that of batch retraining.

1. Introduction

Statistical Machine Translation (SMT) systems largely depend on the availability of parallel corpora for training and tuning. Even more crucial is the availability of sufficient in-domain training data. That is, parallel corpora from the same domain the system will be used for. Methods for dealing with insufficient in-domain data typically fall within the *domain adaptation* line of research. Such methods strive to make the best possible use of the in-domain data or to obtain additional bilingual data that is similar to the target domain. Generally speaking, the more in-domain data is available, the better is the translation.

New training data can become available as more translations are being produced (e.g. in the case of the European Parliament proceedings), through focused data collection, or as the result of user feedback to the translation system. Specifically, *post-editing*, i.e. manual correction of automatic translations, is a useful source for training data.

An additional challenge beyond obtaining sufficient in-domain training data (or any training data), is feeding the new data into an existing up-and-running translation system. A standard way to comprehensively update an SMT model based on new data is to re-train the model with the entire data that is available at a given time. This kind of training is often referred to as *batch* (re-)training. Such a process is time consuming and intensive in computational resources, especially when large datasets are involved. In consequence, it may not be feasible to run it often enough, resulting with long lags between two model batch updates, in which the running system is not up-to-date with the newest possible model.

Incremental training algorithms address this issue by enabling an SMT model update based on the new data rather than retraining the model from scratch. This is performed, for instance, by using online versions of the Expectation Maximization algorithm, that is employed in the alignment step of the SMT model construction. Incremental training for SMT models is a relatively new line of research, and mature tools to perform the required updates efficiently are still largely missing.

Still, as we show in this paper, other configurations of the SMT system are also providing means for utilizing new data in between batch updates. We compare several such configurations, where in-domain data is based on spoken language transcriptions, to assess which methods are practically useful for quickly updating the model, especially when the new data belongs to the target domain.

Consider the following setting of an automatic translation system that is either a standalone translator or as part of a larger software system. The system is deployed and is being used, as more training data is becoming available constantly, e.g. through users who provide corrections to the system's translations. To use this data, two kinds of update cycles are employed: (i) a long cycle (e.g., a week), at which end we can perform a *slow update*, that can include re-training, tuning and any other time-consuming tasks; (ii) a short cycle (a day, for instance) in which we wish to carry out a *quick update* consisting of only light-weight tasks that are guaranteed to complete in a timely manner. In these short cycles the model is updated with the newly obtained data. The goal is to improve the model with respect to the previous slow update, and reflect the received feedback; we do not necessarily expect to obtain as good a performance as the following slow update, but hope to be in the same ballpark. The focus of this work is in identifying the most appropriate setting for quick updates, both in terms of translation quality and of time. That, with tools that are currently available.

In the remaining sections we provide (Section 2) a short background about incremental training and domain adaptation techniques, and discuss the effort of each of the steps in building a phrase-based SMT model; we present the configurations for quick updates that we assessed (Section 3), and describe the experimental setting (Section 4). Section 5 presents the experiments we conducted and their results, and Section 6 summarizes the practical takeaways of this study.

2. Background

2.1. Incremental training for SMT

Incremental training methods provide a principled way for updating an SMT model when more data is received, without re-generating the model from scratch. In addition to efficiency, such methods hold the promise to reflect updates immediately, without work interruption, and are therefore of major importance in many scenarios.

Incremental training for MT often makes use of an online version of the Expectation Maximization (EM) algorithm [1]. EM is used for the purpose of aligning the bilingual corpus while computing translation probabilities [2]. In *Online EM*, the model parameters are updated after each example or a small set of examples (*mini-batch*), and not for the entire dataset at once. Naturally, online EM is faster than *batch EM*, but may be less stable.[1]

Ortiz-Matrínez et al. [4] use incremental online EM [5] to update a standard log-linear model. They apply it in the context of Interactive Machine Translation, where conveying to the user the impression of a highly adaptive system is particularly important. A method for incrementally updating SMT models was also proposed within the SMART project [6]. A large set of features, on top of the standard translation features, is extracted from (simulated) post-edited translations. While the weights of the standard features are tuned offline and remain stable, the weights of the new ones are updated after each source-translation pair. Levenberg et al. [7, 8] use *stepwise EM* for updating the translation model parameters. They use IBM Model 1 [2] with HMM alignments [9], collecting counts for translations and alignments and updating them by interpolating the statistics of the old and the new data. We employ and assess an implementation of this algorithm within Moses (see Section 5.7).

2.2. Domain adaptation

Domain adaptation is the task of adapting a statistical model that was trained on a certain domain to perform well on another domain. Generally, *domain* refers to the distribution of the (training or test) instances; in language-based tasks this term may refer to any of topic, style, dialect, genre or a combination of thereof [10].

Domain adaptation is of major importance for SMT, and in particular for spoken language translation, where bilingual training data is often scarce, and models are thus heavily relying on *out-of-domain* corpora for training. Some methods aim to optimize the use of available corpora through data selection – using only the part of the training data that is more similar to the target domain, or by instance-weighting, i.e. giving each example a weight that corresponds to its similarity to the target domain [11, 12, 13]. In [14], such adaptation in performed on-the-fly without assuming the target domain is known in advance. Other methods apply focused domain-specific data acquisition, e.g. by web crawling [15].

In many scenarios, though, little or no *in-domain* data is accessible in advance. It may be attained at a later stage, e.g. via user feedback to the translation, in the form of post-editing. When such data becomes available, it is desirable to update the model with this data without much delay.

[16] start with an in-domain phrase table, which is then filled-up with new entries from other corpora. In- and out-of-domain entries are distinguished with an additional feature. A more explicit separation of domains is found in the mixture models approach. Training data is divided into components according to the different domains. A model (either a translation model or a language model) is trained for each component separately and the models are then weighted and combined to form a complete model [17, 18]. We use this approach in some of the configurations we assess. However, our goal is different: we focus on the capability to perform the updates quickly. Fortunately, as our results show, these considerations often go hand in hand, and methods that work well for domain adaptation are useful also for quick updates.

2.3. SMT model generation

For completeness, we briefly describe the main steps in generating a basic phrase-based SMT model, from a parallel corpus to a tuned model. Our description corresponds to the steps as done in Moses [19], but is typical to most phrase-based SMT systems.

- **Preprocessing**: The model generation process starts with preprocessing of the bilingual parallel (sentence-aligned) corpus, including tokenization, lower-casing, and removal of sentences that are, e.g., very long.
- **Alignment**: Following some file preparation steps, GIZA++ [20], an implementation of the IBM Models, is applied in two directions (source-target and target-source) to produce word alignment within each source-target sentence pair. A symmetrization of the GIZA bi-directional word alignments follows.
- **Phrase table construction**: Based on word alignments, a *translation model* is generated: lexical (word) translation probabilities are computed and phrases are extracted, scored and stored in a phrase table (PT).
- A **reordering table** is constructed to model position change of phrases between the source and the target.[2]
- A **language model** (LM) is generated from the target side of the parallel corpus and possibly additional target language monolingual data.
- Lastly, **tuning** takes place in order to optimize the weights of individual scores (features) within the complete model.

[1] See [3] for a detailed discussion about the variants of online EM.

[2] The abovementioned phrase extraction is also needed for this step; we chose to include it within the translation model generation step since, as explained later, we do not update the reordering model in this work.

Large phrase tables, reordering tables and language models that cannot fit into memory are often binarized for quick loading and access at translation time. Yet, binarization is not feasible when very large tables are concerned. Reducing the size of the tables through filtering based on a given test dataset is not practical in real world scenarios, and is slow to process as well, as it also depends on the size of the tables.

Of the above, alignment is the most time-consuming step; phrase table construction may also requires a substantial amount of time, especially when binarization is performed; tuning involves multiple iterations (typically over 20) in which a development set is translated and evaluated, and is therefore a highly time-consuming task. Indeed, some of the steps can be parallelized, yet not all. For instance, in MGIZA [21], a multi-threaded version of GIZA++, sentences-pairs are aligned in parallel, saving a substantial amount of time; still, parameter estimation is based on counts that are accumulated from all aligned sentences, and is not parallelized. What is often referred to as *batch training* consists of all the above steps applied to the entire data.

Figure 1 shows the relative time required to complete each task, based on an experiment we conducted, with 1 million sentence-pairs for training and 1,000 sentence-pairs for tuning. Both datasets were taken from the Italian-English corpus of Europarl version 7 [22]. As elapsed time depends on the specific machine and its load at the time of measurement, we use the Unix `time` command for obtaining duration information. We look at the accumulated CPU time, which is roughly equivalent to running on a single CPU. For intuition, the alignment task used up approximately 21 CPU hours, which corresponded to about 6 actual hours when running MGIZA with 4 cores. For comparison, under the same machine configuration, alignment of 2,000 sentences took 2.5 CPU minutes, and 10,000 sentences required less than 13 minutes. This experiment was performed on a 64 bit Linux machine, with four 2.67GHz cores and 50GB of RAM.

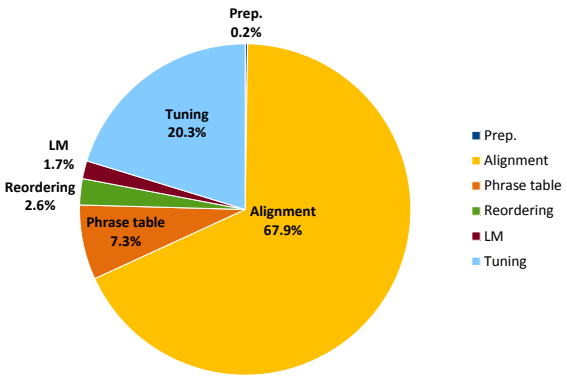


Figure 1: Percentage of the time required by each task of the phrase-based model generation. The times shown here include the binarization of the corresponding model.

3. Quick update configurations

Let's recall the scenario we take interest in: An SMT system is trained based on a large out-of-domain corpus and is meant to be used on a different type of dataset, namely spoken-language texts. The translation service is made available and gradually in-domain data is flowing in. With this data we wish to update the system in the most efficient and effective way. We expect best translations to be produced when we use all the data we have at our disposal; at the same time, we do not wish to carry out intensive processes unnecessarily. We therefore carry out batch updates periodically (long-cycles), and in the interim perform quick, short-cycle updates using the newly obtained data. We wish to identify the most useful configuration – in terms of time and translation quality – for performing such short cycle updates. For that purpose, we examined the following configurations for quick model updates. Each has its pros and cons, as discussed below. Figure 2 depicts their phrase table settings.

1. **OLD-NEW**: In this configuration we use two phrase tables. We maintain all previously obtained ("old") training data, both in-domain and out-of-domain, in one phrase table and the newly obtained data ("new") in a second table. To update the model, we only need to preprocess and align the new data on its own and generate a phrase table from it. This is therefore a very quick way to perform updates.

2. **IN-OUT**: This setting uses two phrase tables as well, but now the out-of-domain data is maintained in one table and the in-domain data in another table. The idea is to allow better model tuning by letting the tuning algorithm give preference to the in-domain table. The drawback is that all in-domain data needs to be processed at every short-cycle update, implying a longer process. As long as in-domain data is limited, this is not an issue. On the contrary, it can contribute to improved alignment quality and phrase table statistics.

3. **3-TABLES**: When in-domain data accumulates, the IN-OUT setting may become too slow. We therefore assess another setting that can potentially combine the benefits of the two above configurations. Here, we use three phrase tables: one for out-of-domain data and two for in-domain. The first among the in-domain tables is used for all previously obtained in-domain data, and the second for the newly obtained data. This way we achieve both separation of in- and out-of-domain data and a quick processing of the new data.

4. **BATCH**: This is a standard setting for phrase-based SMT model generation, used for comparison. The entire training data is concatenated and used together, and a single phrase table is produced. One potential advantage is, as above, an improved alignment quality.

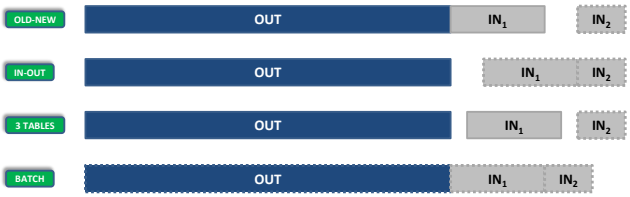


Figure 2: Phrase tables in the different configurations. OUT denotes out-of-domain data; IN_1 is in-domain data previously obtained, and IN_2 is the in-domain data we have just received and wish to use to update the system with. The dashed lines designate the data that needs to be processed in each update cycle.

In addition to the above, we have experimented with incremental updates via Moses' dynamic suffix array. We describe that in Section 5.7.

Required effort Table 1 details which task needs to be performed in each configuration, and the amount of work that has to be done. We explain the required effort of each configuration through an example, whose timeline is presented in Figure 3: We consider a specific point in time of an operational translation system. This system was trained with 1 million out-of-domain sentence-pairs before any in-domain data was available (S_1 in the figure); over time, 30,000 in-domain bi-sentences were received and the system has been already updated with them in a slow update cycle (S_2). Between the previous slow update and the current point in time, 10,000 in-domain sentences have been obtained, and fed into the system (q_{21}); now we receive 5,000 more, and wish to carry out quick update q_{22}.

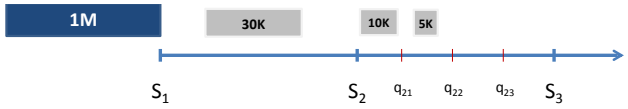


Figure 3: Updates timeline, as described in the example in Section 3. S_i denotes a slow update and q_{ij} a quick update. Boxes represent the available data: dark-shading for out-of-domain and light-shading for in-domain.

Config./ Task	Prep.	Alignment	PT	LM
OLD-NEW	5K	15K	15K	15K
IN-OUT	5K	45K	45K	45K
3-TABLES	5K	15K	15K	15K
BATCH	5K	1,045K	1,045K	1,045K

Table 1: An example of the required effort of each configuration. We consider as "new" all data received since the last slow update.

Here we assume that all data received in between slow updates is small and can be processed together. Preprocessing need not be repeated, but the other steps may perform better given more data. As seen in the table, both OLD-NEW and 3-TABLES require minimal processing. The difference between them is the way the previously-obtained data is stored; IN-OUT requires a more substantial amount of processing, and BATCH requires all the data to be processed from scratch.

So far, our discussion focused on phrase tables. Concerning the LM, Table 1 assumes a setting where the LMs configuration is equivalent to that of the phrase tables. Our experiments showed that – at least for the language pairs we assessed – the reordering model does not significantly affect translation performance; thus, we do not update it in any of the quick update settings. Tuning is discussed in Section 5.6.

4. Setting

4.1. Datasets

We used three datasets in our experiment, two spoken-language parallel corpora, transLectures (TL) and WIT[3] (WIT3 below), and Europarl that represents a large out-of-domain corpus, of non-spoken-language.

transLectures
The first spoken-language dataset was obtained via the transLectures project that is addressing the transcription and translation of scientific video lectures.[3] Translating from English to French, we used the entire datasets that were available at the time of the experiments. These consisted of merely several thousand sentence pairs, that were produced through manual post-editing of the automatic transcription, followed by post-edition of the automatic translation of the transcriptions. The continuous text of the lectures was split into sentences based on long silences in the speech and with a maximal sentence-length constraint. This dataset and its production represent a typical scenario were in-domain spoken language data is scarce, hard to collect and slow to arrive.

- Training set: ∼4,000 English-French sentence pairs.
- Development set: 1,000 bi-sentences, used for tuning.
- Test set: 1,360 sentences-pairs.

WIT3
Our first round of experiments was conducted on the TL data. To confirm the validity of the results across datasets and language pairs, and to allow reproducing our results through a freely available resource, we used another spoken-language dataset, WIT3 [23]. WIT3 (Web Inventory of Transcribed and Translated Talks) is a parallel corpus created from transcription and translation of TED talks.[4] We used a different language pair, Italian to English, and 10-times as much training data as available in the TL dataset.

- Training set: 40,000 sentence-pairs from the Italian-English WIT3 corpus.[5]
- Development set: 1,000 bi-sentences of that corpus.
- Test set: 1,000 bi-sentences from the above corpus.

[3] http://www.translectures.eu/
[4] http://www.ted.com
[5] Downloaded from https://wit3.fbk.eu

Europarl
For each language-pair we used, as part of the training set of most configurations, 1 million Europarl v. 7 bi-sentences.

4.2. Experimental setup

Phrase-based SMT Moses [19] was the translation system used for our experiments. When more than one phrase table was employed, we used the *either* option, meaning that translation options are searched for in either table with no preference to one table over the other, and while not expecting every translation option to be present in both tables.

Alignment Some experiments assessed the use of incremental training and of dynamic suffix arrays. For fair comparison, we used Incremental GIZA [7] in all our experiments rather than GIZA++. However (with the exception of the experiments described in Section 5.7), we did not use its incremental capability.

Language Model We trained 5-gram language models on the target side of the training set(s) using SRILM [24], with modified Kneser-Ney discounting [25].

Tuning Model weights were tuned with batch MIRA [26].

Evaluation We use Smooth (sentence-level) BLEU [27], and report the average score over the test set sentences. All our evaluations were performed on lower case, tokenized texts, using the standard Moses tools for preprocessing.

5. Experiments and results

In this section we present experiments conducted with the TL and WIT3 datasets, and their results.

5.1. Batch updates

We start by providing the results of "regular" batch updates, where the entire training set is used as a single corpus. The first row of each dataset in Table 2 shows the baseline, when no new data is used. This is the starting point of a system that was trained on a large amount of out-of-domain data; in the second row we show the result when 4K (TL) or 40K (WIT3) bi-sentence are used to update the phrase table (i.e. the translation model), but not the LM or the reordering table; the third row shows results of updating all three.

Dataset	Configuration	BLEU
transLectures	Baseline	23.9
	BATCH, PT only	27.9
	BATCH, complete	28.3
WIT3	Baseline	29.4
	BATCH, PT only	30.9
	BATCH, complete	30.7

Table 2: Results of batch updates.

Unsurprisingly, the addition of the new in-domain data to the phrase table greatly improves the translation quality; updating the LM and reordering tables adds a bit more on top of that for transLectures. As mentioned, initial experiments showed that reordering had insignificant impact on results, and improvements may thus be mostly attributed to the LM update; we therefore assessed the performance of all following models without updating the reordering table.

5.2. Quick updates

We now evaluate the performance of quick update models. In these experiments we assume we are about to perform an update equivalent to q_{21} in Figure 3. That is, we have received some in-domain data earlier, performed a slow update since, and now receive additional in-domain data, which we use to quickly update the model. Table 3 shows the results of the three configurations where only the phrase table is being updated with the new data, i.e. the language model and the reordering model are not updated at all. While using the same amount of data as for the batch updates in Table 2, and even with this partial model update, each of these configurations outperforms the batch update, over the two datasets. This result is consistent with prior work on domain adaptation (e.g. [17, 18]), but the important aspect that we are concerned with is that this update is **much** faster. Instead of processing over a million sentence pairs, only up to 4,000 (TL) or 40,000 (WIT3) need to be handled.

Dataset	Configuration	BLEU
transLectures	OLD-NEW	29.4
	IN-OUT	29.7
	3-TABLES	30.2
WIT3	OLD-NEW	31.2
	IN-OUT	31.7
	3-TABLES	31.2

Table 3: Quick updates, where only phrase tables are updated.

5.3. Quick updates of the language model

Next, we evaluate the performance when the LM is also updated. We use multiple LMs, separated the same way as the phrase tables: OLD-NEW and IN-OUT use two LMs, and 3-TABLES, uses three. This allows quick update of this model as well. Table 4 shows the results of this set of experiments.

Dataset	Configuration	BLEU
transLectures	OLD-NEW	31.2
	IN-OUT	31.8
	3-TABLES	31.6
WIT3	OLD-NEW	32.3
	IN-OUT	33.1
	3-TABLES	32.3

Table 4: Quick update results, with matching LM and phrase table configurations.

In all cases, results are improved relative to updating only

the phrase-table (Table 3). Updating the LM was expected to help, yet here we experimentally see that even a quick LM update achieves significant improvements, and is useful for our goal. The best configuration is IN-OUT for both datasets. This is the slowest of the three configurations; hence, depending on the data size, the other options may also be considered, and in particular the 3-TABLES option.

We have seen that quick LM update on top of the phrase table helps; we now wish to verify that updating the LM alone is not sufficient. Table 5 shows two such experiments on the WIT3 dataset. In the first, the target side of the WIT3 training corpus was added to the Europarl corpus to generate a single LM; in the second, the same WIT3 data was used to produce a separate LM. Note that the first among these is not a quick update per-se. Yet, LM generation is much faster than phrase table construction; if the performance is competitive, this can also be an option to consider.

As it turns out, training of a single LM with the additional data did not improve results relative to the baseline. Possibly, in-domain data (consisting of less than 4% or the training data in this case) is diluted in the entire set. More importantly, we see that the quicker update where the LMs are separated, is better. The performance is similar to the configuration where only the phrase table is updated but is inferior to all configurations where both models are updated.

Configuration	**BLEU**
Single LM	29.4
Separate LMs	31.4

Table 5: WIT3, updating only the language model.

5.4. Separating the LMs for batch training

Following the above results where LM separation helps, we assess this option with batch updates as well. Here we maintain a single phrase table, and separate only the LMs. This setting is still slow, yet somewhat quicker than a complete batch update since the previous LM need not be generated, just the new one. The more time-consuming steps of alignment and phrase table construction are still necessary.

Dataset	Configuration	BLEU
TL	BATCH, single LM	28.3
	BATCH, separate LMs	31.6
WIT3	BATCH, single LM	30.7
	BATCH, separate LMs	32.6

Table 6: Comparison of batch configurations, with and without separating the LMs for in/out-of domain data. The single-LM configurations are the same ones shown in Table 2.

Table 6 shows that LM separation significantly improves results also when the PT is batch-trained, and while not considered quick, it is useful to separate the LMs between domains also in this case. The results are still inferior to those obtained by a complete (quick) in-out separation, and are just slightly better than other quick configurations in Table 4.

5.5. No-adaptation

So far our results included two types of datasets. We also wish to understand the effect of the different configurations when only a single domain is concerned. In this setting, IN-OUT and 3-TABLES are not relevant, only OLD-NEW is, with or without phrase table and LM separation. The TL data is too small for this experiment, and we use only WIT3, training a model with $30K$ sentence-pairs and updating it with additional $10K$. The results are shown in Table 7. The first row shows the baseline result before the $10K$ dataset is used, and the second shows the result where all data is trained together in a batch setting. The next two rows show quicker updates: the first – and the quickest – where both phrase table and LMs are separated between old and new data, and the second, where only the phrase tables are separated.

Configuration	BLEU
Baseline	28.2
BATCH	29.2
OLD-NEW	28.5
OLD-NEW, single LM	28.9

Table 7: WIT3 results, where only in-domain data is used.

Now that domain adaptation is no longer a factor, BATCH achieves the best result. Here, we can see the benefit of generating models using the entire data. Quick updates are not far behind, and are faster to carry out. In this setting, separating LMs of the same domain is not useful, and a better model is obtained when more data is used. Notice, though, that these scores are inferior to those obtained in the previous experiments. Out-of-domain data is very useful, and as this is case, quick update methods should still be considered.

5.6. Tuning

Each of the above models was tuned individually before being evaluated. Still, separate experiments show that tuning is not strictly required for every update. Tuning is likely necessary when a configuration is changing, e.g. in terms of components, the data split between them, or the balance between the datasets. When these remain relatively fixed, and a small amount of data is added, tuning may be skipped. Two examples are shown in Table 8. In each, the first row shows the result of a model trained with the Europarl corpus and with partial TL data. The next two models (rows 2 & 3 for each experiment) use additional 1,000 bi-sentences and differ only in the tuning – while the first was re-tuned, the second was not, and used instead the tuned weights of the baseline model. We see that by re-tuning we obtain a small gain in performance; yet, we greatly lose in terms of time. In many cases, then, tuning can be skipped for intermediate updates, and reserved only for slow updates.

Setting	Configuration	BLEU
TL, 2K; IN-OUT	Baseline	27.76
	Re-tuned	28.45
	Not re-tuned	28.31
TL, 4K; OLD-NEW	Baseline	28.51
	Re-tuned	29.37
	Not re-tuned	29.19

Table 8: Tuning with all available data vs. using a model with the same configuration tuned with a smaller amount of data.

So far we have seen several options for model updates that can be applied very quickly. Using the Moses server, once an updated model is ready, it can be loaded into memory practically instantaneously, replacing a previous instance of the server that was loaded with a previous model. That is, as long as all large models are binarized. We can assume binarizing is done during slow updates, and that small models can be loaded quickly and fit into memory easily. With IN-OUT we run into the risk that in-domain data also becomes large; this is not an issue for the 3-TABLES configuration, where the processed data always remains small.

5.7. Incremental training and dynamic suffix arrays

We have extensively experimented with incremental GIZA, and with updates through the dynamic suffix array in Moses.[6] Suffix arrays constitute an alternative to phrase tables, where the entire training data is maintained in memory rather than in a phrase table [28]. Dynamic suffix arrays [7] further enable inserting or deleting training instances, thus updating the translation model without retraining. Although very efficient in comparison to batch training, the process of incrementally updating a model with these tools is not as fast as one would expect. Apart from preprocessing and alignment of the new data (which are required in any case), it requires, prior to the alignment, updating the vocabulary and cooccurrence files, as well as the HMM probabilities. These statistic updates, which operate over the respective files of the entire data, need to be done independently of the size of the new data. It is therefore not efficient to run it per sentence, but rather per mini-batch. Once the new data has been aligned, inserting each bi-sentence into the suffix array is needed to have the translation system updated. Apparently, this is a time consuming process and cannot be considered a real-time update. Creating a phrase table for the new data, and loading another instance of the Moses server, is significantly faster.

We have run multiple comparative experiments with phrase tables vs. suffix arrays, and with combinations of them both, and observed a significant drop in results whenever the suffix array was used, with or without Incremental GIZA. For instance, for the same setting in Table 2, row 1, the BLEU score dropped from 23.9 to 20.8 when the phrase table was replaced with a suffix array. A possible reason is the fact that the inverse translation probabilities are missing in this data structure. Moreover, when an update takes place, the translation server becomes unusable, maintaining the suffix array in memory takes up a large amount of memory and the updated model cannot be saved into disk, but needs to be reconstructed later. Further, updates to the LM are not supported, although this issue was addressed in [8]. All these make this data-structure currently difficult to use or rely on.[7]

The potential advantage of principled incremental training is obvious. Taking into account the previously accumulated data is expected to produce better statistics; doing so while maintaining the system live and constantly updated is a highly sought-after goal. Yet, aligning all data, regardless of the domain, is not always beneficial. Thus, once such tools are stable and efficient, quick updates may be used in conjunction with incremental training and suffix arrays. For instance, out-of-domain data can be maintained in a phrase table, while in-domain data that needs updating is loaded into a suffix array. Preparations for alignment are longer, but the advantage in comparison to IN-OUT is that only alignment of the new data is necessary, but not phrase-table generation.

6. Conclusions

This work focused on identifying simple configurations of phrase-based SMT systems, which allow updating the underlying model quickly when new training data becomes available. We have emphasized the applicability to domain adaption, which is particularly relevant for spoken language applications, where seed in-domain parallel resources are typically scarce or altogether absent. Still, we have shown that this type of updates is suitable also for single-domain settings. We assessed multiple configurations, some of which are based on proven methods from domain adaptation research, to highlight the preferred ones both in terms of translation quality and of processing speed. We described how quick updates can be integrated into the lifecycle of an operational SMT system, enabling efficiently maintaining translation quality while keeping the system up and up-to-date.

Our results show that quick updates are competitive with batch retraining on corpus concatenation, a strong baseline, while being orders of magnitude faster. We have seen that a complete separation of in- and out-of-domain data usually results with best translation quality; yet, this option may become slow over time. The 3-TABLES configuration we proposed solves this issue, albeit at the price of some drop in performance. A potential improvement for this configuration, that we intend to investigate, is to reserve some, moderate size in-domain data for training together with the new data, benefiting from the potential improved alignment, while still keeping the update fast.

[6]We thank Abby Levenberg for his support at this part of the study.

[7]In summer 2013, a new implementation of the dynamic suffix array has been introduced in Moses, where all standard 5 features are computed. Some of the above issues may have been handled. To the best of our knowledge this is still work-in-progress and we have not experimented with it so far.

7. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287755.

8. References

[1] O. Cappé and E. Moulines, "On-line expectation–maximization algorithm for latent data models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 3, pp. 593–613, 2009.

[2] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, June 1993.

[3] P. Liang and D. Klein, "Online em for unsupervised models," in *Proc. of NAACL*, 2009.

[4] D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta, "Online learning for interactive statistical machine translation," in *Proc. of HLT*, 2010.

[5] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. I. Jordan, Ed. MIT Press, 1999, pp. 355–368.

[6] N. Cesa-Bianchi, G. Reverberi, and S. Szedmak, "Online learning algorithms for computer-assisted translation." in *Deliverable D4.2, EU Project SMART*, 2008.

[7] A. Levenberg, C. Callison-Burch, and M. Osborne, "Stream-based translation models for statistical machine translation," in *Proc. of HLT-NAACL*, 2010.

[8] A. Levenberg, "Stream-based statistical machine translation," Ph.D. dissertation, University of Edinburgh, 2011.

[9] S. Vogel, H. Ney, and C. Tillmann, "Hmm-based word alignment in statistical translation," in *Proc. of COLING*, 1996.

[10] B. Chen, R. Kuhn, and G. Foster, "Vector space model for adaptation in statistical machine translation," in *Proc. of ACL*, 2013.

[11] Y. Lu, J. Huang, and Q. Liu, "Improving statistical machine translation performance by training data selection and optimization," in *Proc. of EMNLP-CoNLL*, 2007.

[12] G. F. Foster, C. Goutte, and R. Kuhn, "Discriminative instance weighting for domain adaptation in statistical machine translation," in *EMNLP*, 2010.

[13] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proc. of EMNLP*, 2011.

[14] L. Gong, A. Max, and F. Yvon, "Towards contextual adaptation for any-text translation," in *Proc. of IWSLT*, 2012.

[15] P. Pecina, A. Toral, V. Papavassiliou, P. Prokopidis, and J. van Genabith, "Domain Adaptation of Statistical Machine Translation using Web-Crawled Resources: A Case Study," in *Proc. of EAMT*, 2012.

[16] A. Bisazza, N. Ruiz, and M. Federico, "Fill-up versus interpolation methods for phrase-based SMT adaptation," in *Proc. of IWSLT*, 2011.

[17] G. Foster and R. Kuhn, "Mixture-model adaptation for smt," in *Proc. of WMT*, 2007.

[18] P. Koehn and J. Schroeder, "Experiments in domain adaptation for statistical machine translation," in *Proc. of WMT*, 2007.

[19] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proc. of ACL Demo and Poster Sessions*, 2007.

[20] F. J. Och and H. Ney, "Improved statistical alignment models," in *Proc. of ACL*, 2000.

[21] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Proc. of the ACL Software Engineering, Testing, and Quality Assurance Workshop*, 2008.

[22] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proc. of MT Summit*, 2005.

[23] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proc. of EAMT*, 2012.

[24] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. of Interspeech*, 2002.

[25] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proc. of ACL*, 1996.

[26] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proc. of NAACL-HLT*, 2012.

[27] D. Cer, M. Galley, D. Jurafsky, and C. D. Manning, "Phrasal: A statistical machine translation toolkit for exploring new model features," in *Proc. of the NAACL HLT Demonstration Session*, 2010.

[28] C. Callison-Burch and C. Bannard, "A compact data structure for searchable translation memories," in *Proc. of EAMT*, 2005.

Analyzing the Potential of Source Sentence Reordering in Statistical Machine Translation

Teresa Herrmann, Jochen Weiner, Jan Niehues, Alex Waibel

Institute for Anthropomatics
Karlsruhe Institute of Technology
{teresa.herrmann,jan.niehues,alexander.waibel}@kit.edu, jochen.weiner@student.kit.edu

Abstract

We analyze the performance of source sentence reordering, a common reordering approach, using oracle experiments on German-English and English-German translation. First, we show that the potential of this approach is very promising. Compared to a monotone translation, the optimally reordered source sentence leads to improvements of up to 4.6 and 6.2 BLEU points, depending on the language. Furthermore, we perform a detailed evaluation of the different aspects of the approach. We analyze the impact of the restriction of the search space by reordering lattices and we can show that using more complex rule types for reordering results in better approximation of the optimally reordered source. However, a gap of about 3 to 3.8 BLEU points remains, presenting a promising perspective for research on extending the search space through better reordering rules. When evaluating the ranking of different reordering variants, the results reveal that the search for the best path in the lattice performs very well for German-English translation. For English-German translation there is potential for an improvement of up to 1.4 BLEU points through a better ranking of the different reordering possibilities in the reordering lattice.

1. Introduction

The reordering problem is commonly acknowledged to be one of the main difficulties in machine translation. One widely used approach is to perform reordering as a preprocessing step before translation. The idea is to synthesize a sentence in the source language that simulates the word order of the target language. Reordering the source text results either in a deterministically reordered sentence or multiple reordering variants are generated and stored in a lattice. Then monotone translation can be performed either on the reordered source sentence or the machine translation decoder searches for the best sequence of words in the reordering lattice.

We want to assess the benefits of this common approach of reordering the source before translation and investigate whether it really helps improve the translation quality. For one, we want to determine lower and upper bounds for the translation quality that can be reached by this approach and to identify potential of further development. Furthermore, we want to assess the performance of the reordering model on two levels: The restriction of the search space of possible reorderings and the ranking of different reordering variants.

We designed oracle experiments that address the following questions:

- How good is the translation of the optimally reordered source sentence?
- How beneficial is the restriction of the search space through reordering lattices for translation quality?
- How accurate is the search for the best path in the reordering lattice?

The paper is structured as follows: First, we present related work dealing with the reordering problem, mainly focussing on reordering as preprocessing and the judgement of reordering quality. In Section 3 we explain the reordering approaches applied in this work in detail. Then we describe the setup for the oracle experiments, which include an oracle reordering of the source sentence and the oracle path in the input lattices which is closest to the oracle reordering. We show the results of the experiments in Section 5 and then draw conclusions about future development of the reordering approach in the final section.

2. Related Work

In our work we investigate the benefits of a pre-reordering approach for machine translation by performing oracle experiments. We first present related work regarding reordering methods in machine translation and reference work on judging the quality of a given reordering. Then we mention work using oracles for the analysis of machine translation systems.

Word reordering has been addressed by many approaches in statistical systems. In a state-of-the-art phrase-based machine translation system, the decoder processes the source sentence left to right, but allows changes in the order of source words while the translation hypothesis is generated. Many phrase-based systems also include a lexicalized reordering model [1] which provides additional reordering information for phrase pairs. It stores statistics on the orientation of adjacent phrase pairs on the lexical level.

A very popular approach is to detach the reordering from the decoding procedure and to perform the reordering on the source sentence before translation. Such pre-reordering approaches use linguistic information about the source and or target language, such as parts-of-speech, dependency or constituency tree structure. They apply hand-crafted rules or automatically learn rules that change the order of the source sentence. Then monotone translation is performed.

In the first pre-reordering approach, reordering rules for English-French translation are automatically learned from source and target language dependency trees [2]. Since then many adopted this method. In the beginning manually crafted reordering rules based on syntactic or dependency parse trees or part-of-speech tags were designed for particular languages [3, 4, 5, 6]. Later data-driven methods followed, learning reordering rules automatically based on part-of-speech tags or syntactic chunks [7, 8, 9, 10]. Alternatively, word class information may be used to perform a translation of the original source sentence into a reordered source sentence [11]. More recent work includes reordering rules learned from source and target side syntax trees [12], automatically learned reordering rules from IBM1 alignments and source side dependency trees [13] and using a classifier to predict source-sentence reordering [14]. An approach presenting automatically learned reordering rules based on syntactic parse tree constituents [15] further combines the tree-based rules with two types of part-of-speech-based rules [7, 10]. This produces complementary reordering variants which result in an improved translation quality. While some of the presented approaches perform a deterministic reordering of the source sentence, others store reordering variants in a word lattice leaving the selection of the reordering path to the decoder.

Related work regarding reordering metrics and reordering quality includes the first description of reorderings as permutations [16]. Later, the use of permutation distance metrics to measure reordering quality [17] leveraged research into distance functions for ordered encodings. An approach to transform alignments into permutations [18] takes the particular characteristics of alignment functions into account.

Oracle experiments have shown to be a valuable method for analyzing different aspects of machine translation. While an oracle BLEU score may serve for identifying translation errors in the phrase table [19], another approach uses oracles for punctuation and segmentation prediction in speech translation [20]. Efficient methods for finding the best translation hypothesis in a decoding lattice have been proposed [21]. Furthermore, research on oracles regarding the reordering problem have been conducted [22, 23]. The first uses linear programming to compare the best achievable BLEU scores when using different reordering constraints [22]. The latter presents a reordering method for translations from English to Spanish, Dutch and Chinese where deterministic reordering decisions are conditioned on source tree features and compared to several oracles [23].

Rule Type	Example Rule	
Short	*VVIMP VMFIN PPER*	$\rightarrow$ 2 1 0
Long	*VAFIN * VVPP*	$\rightarrow$ 0 2 1
Tree	*VP PTNEG NP VVPP*	$\rightarrow$ 0 2 1

Figure 1: *Rule Types*

Our work differs in three ways: First, we investigate a reordering approach where reordering decisions are not deterministic. Instead, reordering variants produced by both part-of-speech-based and tree-based reordering rules are stored in a lattice and the final order of the source sentence is decided during decoding. Second, we perform a separate analysis of two different aspects: the quality of the restriction of the search space through reordering lattices and the accuracy of the search. Third, we perform translations from English to German and German to English for 2 different translation tasks.

3. Reordering Approach

We first describe the reordering methods applied in the systems used in our oracle experiments. We use two approaches based on continuous and discontinuous sequences of parts-of-speech of the words in the sentence [7, 10]. In addition we perform reordering based on constituents of syntactic parse trees [15] and we combine the different types of rules. Thus, we cover both short-range and long-range reordering phenomena between source and target language.

3.1. Rule Types

In our experiments we distinguish between short-range, long-range and tree-based rules. Examples for each of the rule types are presented in Figure 1.

3.1.1. Short-range Rules

Short-range rules consist of a sequence of part-of-speech (POS) tags on the left hand side and an indexed representation of the target order of those POS tags on the right hand side of the rule. Each rule comes with an associated probability which is the relative frequency of the occurrence of this reordering in the training corpus.

3.1.2. Long-range Rules

A long-range rule consists of a sequence of POS tags with placeholders on the left hand side. Placeholders can match arbitrary types and numbers of POS tags. The right hand side of the rule contains the reordered indices where the tags matched by the placeholder are assigned one index as a whole. Again, a probability is assigned to each rule.

3.1.3. Tree-based Rules

The tree-based rules address reordering within one constituent of a syntactic tree. The rule consists of the head category as well as the the child categories of the constituent on the left hand side of the rule. The right hand side represents the reordered sequence of the children where each child constituent is assigned one index and the words covered by it are moved as a whole. An additional type of rules called partial rules need not cover all the children in the constituent, but consecutive sequences of children.

3.2. Learning Reordering Rules

For the training of the reordering rules a parallel corpus and a word alignment is required. In addition, we need the POS tags for the source side of the corpus for training the POS-based reordering rules. For the tree-based rules we need syntactic parse trees for the source side. For each sentence in the training corpus we search for changes of word order between the source and target language sentence. When we find a crossing alignment indicating a different order of source and target language words, we monotonize the alignment and extract a rule that rearranges the source words in the order of the aligned target words. For more details refer to the descriptions of POS-based rules [7, 10] and tree-based rules [15].

3.3. Applying Reordering Rules

Before translation, a word lattice is created that includes the original source sentence as the monotone translation path. Initially all edges of the monotone path are assigned a transition probability of 1. Then the reordering rules are applied to the source text. For each sentence all applicable rules are applied where the tree rules might be applied recursively to reordered paths. The resulting reordering variants are stored in the word lattice. The edges of the reordered path are assigned transition probabilities according to the probability of the applied reordering rule. An edge branching from the monotone path receives the probability of the rule. The following edges in the reordered path are assigned a probability of 1. The edge on the monotone path where the branching started receives an update such that the probability of the applied rule is subtracted from the current transition probability of this edge. Finally, the word lattice including all reordering variants is used as input to the decoder.

3.4. Judging Reordered Paths

The probability of a given path in a reordering lattice is calculated as the product of the individual transition probabilities of the traversed edges. Since the transition probabilities are based on the occurrences of the reordering in the training data, the highest scoring path in the lattice should represent the best reordering for the sentence. The reordering lattice is one model in the log-linear model combination of the translation system. Its weight is set during optimization of the whole system together with the weights of the other models in the translation system.

4. Oracle Reordering

We want to investigate the impact of the reordering on the translation quality. We compare the actual system performance against two different oracle reorderings of the input sentence. With these experiments we want to address the questions raised in the introduction.

The first oracle is the optimally reordered source sentence which presents the source words according to the target language word order. With this experiment we analyze the usefulness of the pre-reordering approach. By reordering the source sentence according to the target language word order we estimate an upper bound for translation quality using this strategy.

Then we investigate how the reordering lattices produced by our reordering model restrict the search space for translation. Therefore, we compare the aforementioned oracle translation with the translation of the oracle path. It corresponds to the path in the lattice that is closest to the oracle reordering of the source sentence. We perform this experiment for each of the different rule types.

In a third experiment we evaluate how good our models are at determining the best path in the lattice. In order to evaluate this aspect, we compare the translation of the oracle path with the actual translation.

4.1. Optimally Reordered Sentence

In order to measure the oracle performance of the pre-reordering approach, we use an optimally reordered sentence as input to the translation system and do not allow additional reordering during decoding. In order to create this oracle reordering for the source sentence, we make use of the word alignment between source sentence and reference translation. This alignment is generated by applying the alignment model trained during system development to the test data and its reference translation. After source and reference are aligned, we create a permutation of the source sentence [17].

In the permutation, words are generally assigned the position of word they are aligned with. However, permutations are one-to-one alignments, while word alignments may also contain unaligned words, many-to-one alignments and one-to-many alignments. Therefore, some simplifying assumptions have to be made when transforming alignments to permutations [18]: *unaligned source words* are aligned to the word after its predecessor or to the first word if it has no predecessor; *unaligned target words* are irrelevant to the source sentence order and are therefore ignored; for *many-to-one source-to-target alignments* the ordering is assumed to be monotone; in *one-to-many source-to-target alignments* the word is assumed to be aligned to the first target word. We will refer to this reordered source sentence as the oracle reordering of the input sentence.

4.2. Oracle Path

With our reordering model we generate many reordering variants by applying reordering rules to the source sentence and store these variants in a lattice. In order to know the upper bound of the restriction of the search space by the lattice we want to identify the best reordering variant in the reordering lattice. We define it as the path in the lattice which has the smallest distance to the oracle reordering as described above.

Among Hamming distance, Ulam's Distance and Kendall's tau distance, a version of Kendall's tau resulted to be the best distance, being the most reliable and correlating strongly with human fluency judgement [17]. Hence, we calculate the Kendall's tau distance [24] in order to find the path that is closest to the oracle reordering. The Kendall's tau distance is the minimum number of swaps between two adjacent symbols that transforms a permutation σ into another permutation π. This metric measures relative differences and takes both the number and the size of reorderings into account. We use the square root version [18] which corresponds closely with human perception of word order quality:

$$d(\pi, \sigma) = 1 - \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=i}^{n} x_{ij}}{Z}}$$

$$\text{where } x_{ij} = \begin{cases} 1 & \text{if } \pi(i) < \pi(j) \text{ and } \sigma(i) > \sigma(j) \\ 0 & \text{otherwise} \end{cases}$$

$$\text{and } Z = \frac{n \cdot (n-1)}{2}$$

If a path with the oracle reordering is in the lattice, this path is the closest path. However, if the oracle reordering is not in the lattice, several paths can have the smallest distance to the oracle reordering. Then we create lattices containing only the best paths and use these as input to the translation system.

Note that the best path or even the oracle reordering need not result in the best possible translation quality for two reasons. First, we rely on the alignment between source and reference for generating the oracle reordering. Errors in the alignment can introduce errors into the oracle reordering and the closest path. Another reason is that we generate an artificial word order which does not match the word order as seen in the training data. Therefore, we might not have well matching phrase pairs for generating the best possible translation.

5. Experiments

In this section we present three experiments designed to address the three questions raised in the introduction. First, we will briefly describe the systems we used to generate the translations. Afterwards, we will analyze the potential of the pre-reordering approach. Then we investigate how the reordering lattices produced by our reordering model restrict the search space for translation. In a third experiment we compare the oracles with the actual performance of a system using the reordering lattices to see how good our models are at ranking different word orders.

5.1. System Description

We perform experiments with four different systems covering two translation directions and two different translation tasks. We translate between German and English in both directions. For each direction we use competitive systems used in WMT and IWSLT evaluations to translate News texts and TED talks in order to cover different domains. For the News systems, the training data includes the European Parliamentary Proceedings and the News Commentary data. The test data is news2011. For details of the WMT system refer to the WMT system description [25]. The systems are optimized once on news2010, but in the experiments described in this paper, no new optimizations were run between system variants using different rule types to reduce the noise to a minimum. The system translating TED talks is trained on European Parliamentary Proceedings, News Commentary data, the Common Crawl corpus and TED talks, while development and test data consist of TED talks only. Again, the systems are only optimized once. A detailed system description can be referred to in [26]. All translations are produced using the input sentence with a word order stated in the given experiment description. No additional reordering in the decoder is allowed.

5.2. Potential of Reordering the Source Sentence

When applying reordering as preprocessing, it is commonly assumed that arranging the source sentence according to target language word order should result in better translation quality. We want to question this assumption and investigate the benefits of the pre-reordering approach in this first experiment that identifies the lower and upper bounds of translation quality with respect to word order. We consider the lower bound of translation quality to be the performance that is obtained by translating the monotone source sentence without allowing any additional reordering. Since the objective of the pre-reordering approach is to obtain the source words in the order of the target language words, we regard the translation of the optimally reordered path to be the upper bound for translation quality. We generate the optimally reordered path using the reference translation and the alignment between source and reference as described in Section 4.1.

5.2.1. German-English

Table 1 presents the results for the translation from German to English in two different domains. The difference between monotone translation and the translation of the oracle reordering is 5.2 and 6.2 BLEU points, respectively. With a system using our lattice-based reordering approach that does not have any oracle information, but the decoder chooses the path, we achieve a performance that is approximately in the middle of that range.

Reordering Type	News	TED
Monotone	20.23	27.18
Lattice Reordering	22.45	30.87
Oracle	25.42	33.39

Table 1: *Oracle Reordering: German-English*

5.2.2. English-German

For the other translation direction, we can see lower absolute BLEU scores, since translation into German is more difficult due to the highly inflective morphology of the German language. Compared to German-English translation, the difference between monotone and oracle translation is smaller, 2.9 and 4.6 BLEU points, respectively. The decoder using lattice reordering performs better than the monotone translation, but the gap towards the oracle translation is bigger. That means that for English to German translation, there is even more potential for improvement through better reordering lattices.

Reordering Type	News	TED
Monotone	15.91	24.22
Lattice Reordering	16.34	24.95
Oracle	18.84	28.77

Table 2: *Oracle Reordering: English-German*

From this experiment we can draw the conclusion that reordering the source text prior to translation indeed holds promising results. Our system using reordering lattices as translation input outperforms the monotone translation in all four translation tasks, and the oracle reordering shows that there is still potential for improvement through better reordering methods. In the following we will investigate how we can best address this potential by analyzing different aspects of the reordering approach in detail.

5.3. Lattice-based Restriction of the Search Space

In the previous experiment we have identified a gap between the actual performance of the system using reordering lattices and the oracle reordered translation. In our reordering approach we restrict the search space of possible reorderings by the reordering lattice. In this second experiment we want to investigate how much this restriction influences the drop in performance. Therefore, we evaluate how much better we could get, if the decoder found the best path in the given reordering lattices. As described in Section 4.2 we define the best path as the one that is closest to the oracle reordered sentence used in the previous experiment.

In order to compare the benefits of individual reordering rule types we apply all the different types of reordering rules and identify the oracle path within the lattices produced by those rules. Then we perform translation of the oracle path and compare the translation quality.

All results tables repeat the scores for the monotone and oracle translation presented above. In addition, they show the translation results for systems using first short and long-range rules based on POS tags. Afterwards follow the tree-based rules, first the plain tree rules, then the tree-based rules with recursive rule application and the third tree rule option includes partial rules. More details on recursive rule application and partial rules are described in [15]. The three final systems combine all rule types.

5.3.1. German-English

Table 3 shows the results for German-to-English translation and the size of the search space by indicating the number of edges in the lattices. As can be seen, the more complex the rule types that are used to generate the reordering lattice and the larger the search space gets, the better the translation of the oracle path in that lattice. Hence, we are able to improve the word order by increasing the search space. The oracle path that is closest to the oracle reordering stems from the lattice produced by applying all rule types.

Reordering Type	News		TED	
	BLEU	Size	BLEU	Size
Monotone	20.23		27.18	
Short	21.37	193K	29.98	68K
Short+Long	21.41	255K	30.66	163K
Tree	21.88	140K	29.74	51K
Tree-rec	22.17	244K	30.11	81K
Tree-rec-partial	22.28	249K	30.22	82K
Short+Long+Tree	22.49	429K	30.97	182K
Short+Long+Tree-rec	22.64	534K	31.10	212K
Short+Long+Tree-rec-part.	22.65	538K	31.12	213K
Oracle	25.42		33.39	

Table 3: *Oracle Path: German-English*

5.3.2. English-German

Table 4 presents the same experiments for English-to-German translation. Again, the more complex rules and bigger search spaces lead to better oracle paths.

Thus, we can confirm the findings in [15], namely that the different rule types produce complementary reordering possibilities which result in the best translation quality if combined in one lattice. We can also see that the translation of the best oracle path is still far from the oracle reordered translation. The lattices generated with the help of our reordering rules restrict the search space in a sensible way to allow for reorderings that are getting closer to the oracle reordered sentence. However, some reordering possibilities are still missing from our lattices. Therefore, research in the area of extending the search space by better rules seems to be promising.

Reordering Type	News				TED			
	DecoderPath		OraclePath		DecoderPath		OraclePath	
	BLEU	Distance	BLEU	Distance	BLEU	Distance	BLEU	Distance
Monotone			20.23				27.18	
Short	21.59	0.290	21.37	0.250	30.00	0.179	29.98	0.124
Long	21.35	0.286	21.41	0.259	30.73	0.181	30.66	0.112
Tree	21.78	0.286	21.88	0.250	29.60	0.180	29.74	0.140
Tree-rec	22.01	0.284	22.17	0.243	29.88	0.179	30.11	0.135
Tree-rec-partial	22.10	0.284	22.28	0.241	29.96	0.179	30.22	0.133
Short+Long+Tree	22.33	0.289	22.49	0.224	30.82	0.182	30.97	0.106
Short+Long+Tree-rec	22.44	0.288	22.64	0.220	30.86	0.182	31.10	0.104
Short+Long+Tree-rec-partial	22.45	0.288	22.65	0.220	30.87	0.182	31.12	0.104
Oracle			25.42				33.39	

Table 5: *Oracle vs. Real: German-English*

Reordering Type	News		TED	
	BLEU	Size	BLEU	Size
Monotone	15.91		24.22	
Short	16.31	186K	25.83	76K
Short+Long	16.70	383K	25.99	170K
Tree	16.48	189K	25.31	71K
Tree-rec	16.60	726K	25.49	237K
Tree-rec-partial	16.60	727K	25.49	237K
Short+Long+Tree	17.00	496K	26.28	208K
Short+Long+Tree-rec	17.07	1M	26.38	373K
Short+Long+Tree-rec-part.	17.07	1M	26.38	373K
Oracle	18.84		28.77	

Table 4: *Oracle Path: English-German*

5.4. Ranking different word orders

The experiments above revealed the best translation that can be produced by using the individual rule types and combinations thereof. Now we want to examine how well we actually perform in finding the best path in the lattices. Again, we tested on all the different rule types, but let the decoder find the best path for translation. It is worth mentioning that the decoder does not only utilize the reordering model described in Section 3 to find the path, but all the models in the log-linear model of the translation system. For reference we include the scores achieved with the oracle paths from the previous experiment. In addition, we present the average distances between the decoder path used for translation and the optimally reordered sentence both for the decoder translation and for the translation of the oracle path. The distances are calculated using the Kendall's tau metric.

5.4.1. German-English

We present the results for German-to-English translation in Table 5. The differences between the oracle path scores and the real performance of the system (decoder path) with the reordering lattices are actually very small. This means that the decoder is already quite good at finding the best path in the reordering lattice. To reach the translation quality of the oracle path, a further increase of 0.2 and 0.3 BLEU points would be possible for the News and the TED task, respectively.

The distances between decoder translation path and oracle reordering are shown in the column to the right of the decoder path, while the distances between the oracle path and the oracle reordering are shown in the column to the right of the scores reached by the oracle path translations. We can see that both the distances and the translation quality for the oracle path systems converge nicely for the News task. The closer the translation quality comes to the translation quality of the oracle reordering, the smaller the distance to the oracle reordering. In the TED task we also observe a good correspondence between translation quality and reordering distance for the oracle path results. The drop in BLEU score when using only tree rules is also obvious in the distance scores, which raise for those systems. For the decoder translation path, the distance to the oracle reordering seems to be not converging at all, it stays about the same both for News and TED translations.

5.4.2. English-German

The results for English-to-German translation are presented in Table 6. For this translation direction, the path in the reordering lattices chosen by the decoder is not very close to the optimal one yet. The decoder performance is 0.7 BLEU points worse than the translation of the oracle path in the best rule type of the News task. For the TED task, the difference between oracle path translation and decoder performance is even 1.4 BLEU points.

The distance scores show a similar behavior as observed in the other translation direction. The distances from oracle path to oracle reordering get smaller as the translation quality increases. The distances from decoder translation path to oracle reordering do not converge. Compared to the other

Reordering Type	News				TED			
	DecoderPath		OraclePath		DecoderPath		OraclePath	
	BLEU	Distance	BLEU	Distance	BLEU	Distance	BLEU	Distance
Monotone			15.91				24.22	
Short	16.27	0.297	16.31	0.249	24.83	0.200	25.83	0.141
Long	16.31	0.311	16.70	0.236	24.87	0.214	25.99	0.129
Tree	16.21	0.306	16.48	0.252	24.47	0.206	25.31	0.163
Tree-rec	16.18	0.312	16.60	0.244	24.51	0.207	25.49	0.158
Tree-rec-partial	16.18	0.312	16.60	0.244	24.50	0.207	25.49	0.158
Short+Long+Tree	16.32	0.318	17.00	0.227	24.94	0.217	26.28	0.123
Short+Long+Tree-rec	16.34	0.321	17.07	0.222	24.95	0.218	26.38	0.120
Short+Long+Tree-rec-partial	16.34	0.321	17.07	0.222	24.95	0.218	26.38	0.120
Oracle			18.84				28.77	

Table 6: *Oracle vs. Real: English-German*

direction they vary even more. It is possible that this is due to the smaller differences in translation quality. In addition, outliers in the paths chosen by the decoder could cause the variations in the distance scores.

From these results on the translation quality we can draw the conclusion that there still lies some potential in the reordering rules and consequently in the reordering lattices that the decoder is not yet able to make use of. The differences in the decoder path translation scores and oracle path translation scores suggest that more complex scoring models for better assessing the quality of different reordering possibilities seem to be a promising research direction for English-German translation.

6. Conclusion

We have analyzed the performance of an approach to reordering as a preprocessing step using oracle experiments. We conducted experiments on German-to-English and English-to-German translation of News texts and TED talks.

In a first series of experiments we could show that source sentence reordering is a very promising approach. By translating an optimally reordered source sentence, we could improve the translation performance by up to 6.2 BLEU points.

Then we translated the optimally reordered source sentence and compared it with the oracle path in reordering lattices produced by different types of reordering rules. This led to the conclusion that the restriction of the search space using our reordering lattices approximates the oracle reordering better when more complex and complementary reordering rules are used. However, the best oracle path and the oracle reordering are still far apart, leaving a lot of potential for finding better reordering rules that approximate the oracle reordering even better. While for German-to-English translation the distance between actual performance and the best possible translation is 2.5 to 3 BLEU points, the gap for English-German is a little bigger. An additional 2.5 to 3.8 BLEU points are missing until the best possible translation result can be reached. As a consequence, one direction of promising research is to extend the search space further to include reordering variants that better approximate the optimally reordered source sentence.

Comparing the decoder path translation with the oracle path showed that the path chosen by the decoder is quite close to the oracle path, both in terms of translation quality and reordering distance for German-to-English translation. The decoder translation path and the oracle path are only 0.2 and 0.3 BLEU points apart. Consequently, the current models used in the machine translation system are able to find almost the best source word order that is in the search space. For English-to-German translation, however, finding the best path in the reordering lattice seems to be more difficult. A gap of 0.7 and 1.4 BLEU remains until the oracle path performance is reached. We can conclude that at least for English-to-German translation a better ranking of the different reordering possibilities in the search space seems to hold a promising perspective for future research.

All in all, our experiments confirmed the usefulness of reordering the source sentence before translation. The approach displayed a good performance with potential for improvement by extending the search space of reordering possibilities. For English-to-German the ranking of reordering quality for finding a better path in the reordering lattice is another promising research direction. In total, the approach has a potential for a further 3 and 3.8 BLEU points of improvements, depending on the language. This potential could be reached by improving the restriction of the search space with better rules and a better ranking of reordering quality.

7. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

8. References

[1] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh system description for the 2005 IWSLT speech translation evaluation," in *Proceedings of IWSLT 2005*, Pittsburgh, USA, 2005.

[2] F. Xia and M. McCord, "Improving a statistical MT system with automatically learned rewrite patterns," in *Proceedings of COLING 2004*, Geneva, Switzerland, 2004.

[3] M. Collins, P. Koehn, and I. Kučerová, "Clause Restructuring for Statistical Machine Translation," in *Proc. of ACL 2005*, Ann Arbor, Michigan, USA, 2005.

[4] M. Popović and H. Ney, "POS-based Word Reorderings for Statistical Machine Translation," in *Proceedings of LREC 2006*, Genoa, Italy, 2006.

[5] N. Habash, "Syntactic preprocessing for statistical machine translation," *Proceedings of MT Summit*, 2007.

[6] C. Wang, M. Collins, and P. Koehn, "Chinese syntactic reordering for statistical machine translation," in *Proceedings of EMNLP 2007*, Prague, Czech Republic, 2007.

[7] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *TMI*, Skövde, Sweden, 2007.

[8] Y. Zhang, R. Zens, and H. Ney, "Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation," in *Proceedings of SSST 2007*, Rochester, NY, USA, 2007.

[9] J. M. Crego and N. Habash, "Using Shallow Syntax Information to Improve Word Alignment and Reordering for SMT," in *Proceedings of ACL-HLT 2008*, Columbus, Ohio, USA, 2008.

[10] J. Niehues and M. Kolss, "A POS-Based Model for Long-Range Reorderings in SMT," in *Proceedings of WMT 2009*, Athens, Greece, 2009.

[11] M. R. Costa-jussà and J. A. R. Fonollosa, "Statistical Machine Reordering," in *Proceedings of EMNLP 2006*, Sydney, Australia, 2006.

[12] M. Khalilov, J. Fonollosa, and M. Dras, "A new subtree-transfer approach to syntax-based reordering for statistical machine translation," in *Proceedings of EAMT 2009*, Barcelona, Spain, 2009.

[13] D. Genzel, "Automatically learning source-side reordering rules for large scale machine translation," in *Proceedings of COLING 2010*, Beijing, China, 2010.

[14] U. Lerner and S. Petrov, "Source-side classifier preordering for machine translation," in *Proceedings of EMNLP 2013*, Seattle, Washington, USA, 2013.

[15] T. Herrmann, J. Niehues, and A. Waibel, "Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation," in *Proceedings of SSST 2013*, Altanta, Georgia, USA, 2013.

[16] J. Eisner and R. W. Tromble, "Local Search with Very Large-Scale Neighborhoods for Optimal Permutations in Machine Translation," in *Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, 2006.

[17] A. Birch, M. Osborne, and P. Blunsom, "Metrics for MT Evaluation: Evaluating Reordering," *Machine Translation*, vol. 24, no. 1, 2010.

[18] A. Birch, "Reordering Metrics for Statistical Machine Translation," Ph.D. dissertation, University of Edinburgh, 2011.

[19] G. Wisniewski, A. Allauzen, and F. Yvon, "Assessing phrase-based translation models with oracle decoding," in *Proceedings of EMNLP 2010*, Cambridge, Massachusetts, USA, 2010.

[20] E. Cho, J. Niehues, and A. Waibel, "Segmentation and Punctuation Prediction in Speech Language Translation Using a Monolingual Translation System," in *Proceedings of IWSLT 2012*, Hong Kong, 2012.

[21] A. Sokolov, G. Wisniewski, and F. Yvon, "Computing lattice BLEU oracle scores for machine translation," in *Proceedings of EACL 2012*, Avignon, France, 2012.

[22] M. Dreyer, K. Hall, and S. Khudanpur, "Comparing Reordering Constraints for SMT Using Efficient BLEU Oracle Computation." in *Proc. of SSST 2007*, Rochester, USA, 2007.

[23] M. Khalilov and K. Sima'an, "Context-sensitive syntactic source-reordering by statistical transduction," in *Proceedings of IJCNLP 2011*, Chiang Mai, Thailand, 2011.

[24] M. Kendall and J. D. Gibbons, *Rank Correlation Methods*, 5th ed. A Charles Griffin Title, September 1990.

[25] J. Niehues, Y. Zhang, M. Mediani, T. Herrmann, E. Cho, and A. Waibel, "The Karlsruhe Institute of Technology Translation Systems for the WMT 2012," in *Proceedings of WMT 2012*, Montreal, Canada, 2012.

[26] T.-L. Ha, J. Niehues, T. Herrmann, M. Mediani, E. Cho, Y. Zhang, I. Slawik, and A. Waibel, "The KIT Translation Systems for IWSLT 2013," in *Proceedings of IWSLT 2013*, Heidelberg, Germany, 2013.

CRF-based Disfluency Detection using Semantic Features for German to English Spoken Language Translation

Eunah Cho, Thanh-Le Ha, Alex Waibel

International Center for Advanced Communication Technologies - InterACT
Institute of Anthropomatics
Karlsruhe Institute of Technology, Germany
{eunah.cho|thanh-le.ha|alex.waibel}@kit.edu

Abstract

Disfluencies in speech pose severe difficulties in machine translation of spontaneous speech. This paper presents our conditional random field (CRF)-based speech disfluency detection system developed on German to improve spoken language translation performance.

In order to detect speech disfluencies considering syntactics and semantics of speech utterances, we carried out a CRF-based approach using information learned from the word representation and the phrase table used for machine translation. The word representation is gained using recurrent neural networks and projected words are clustered using the k-means algorithm. Using the output from the model trained with the word representations and phrase table information, we achieve an improvement of 1.96 BLEU points on the lecture test set. By keeping or removing human-annotated disfluencies, we show an upper bound and lower bound of translation quality. In an oracle experiment we gain 3.16 BLEU points of improvement on the lecture test set, compared to the same set with all disfluencies.

1. Introduction

Natural language processing (NLP) tasks often suffer from disfluencies in spontaneous speech. In spontaneous speech, speakers occasionally talk with disfluencies such as repetitions, stuttering, or filler words. These speech disfluencies inhibit proper processing for other subsequent applications, for example machine translation (MT) systems.

MT systems are generally trained using well-structured, cleanly written texts. The mismatch between this training data and the actual test data, in this case spontaneous speech, causes a performance drop. A system which reconstructs the non-fluent output from an automatic speech recognition (ASR) system into the proper form for subsequent applications will increase the performance of the application.

A considerable number of works on this task such as [1] and [2] focus on English, from the point of view of the ASR systems. One of our goals is to extend this work to German, and also apply it to the MT task, in order to analyze the effect of speech disfluencies on MT.

1.1. Disfluencies in Spontaneous Speech

Filler words (e.g. "uh", "uhm") are a common disfluencies in spontaneous speech. Discourse markers (e.g. "you know", "well" in English) are considered filler words as well. Another common disfluency is repetition, where speakers repeat their words. A repetition can either be an identical repetition, where speakers exactly repeat a word or phrase, or a rough repetition, where they correct themselves using similar words. Simplified examples of such repetitions from our disfluency annotated lecture data with English gloss translation are shown in Table 1, in which the identical repetition is on the upper part, and the rough repetition is on the lower part.

Table 1: *Repetitions in spontaneous speech*

Source	Das sind die Vorteile, **die Sie die Sie** haben.
En.gls	These are the advantages, **that you that you** have.
Source	**Da gibt es da gab es** nur eins.
En.gls	**There is there was** only one.

Another type of speech disfluency, where several speech fragments are dropped and new fragments are introduced, is restart fragments. As presented in Table 2, the speaker starts a new way of forming the sentence after aborting the first several utterances. Although the example shown in this table depicts a case where the context is still kept in the following new utterances, occasionally we confront other cases where the previous context is abandoned and a new topic is discussed in spontaneous speech.

Table 2: *Restart fragment in spontaneous speech*

Source	**Das ist alles, was Sie** das haben Sie alles gelernt, und jetzt können Sie...
Engl. gloss	**That is all, what you** you have learned all of this, and now can you...

1.2. Motivation

Detecting obvious filler words and simple repetitions can be more feasible than other sorts of disfluencies for automatic modeling techniques, using lexical patterns such as typical filler word tokens and repetitive part-of-speech (POS) tokens as in previous work [2, 3]. Although it is the case for obvious disfluencies (i.e. "uh", "uhm", same repetitive tokens, and so on), we are confronted with many other cases where it is hard to recognize or decide whether the token is a disfluency or not via automatic means. This issue can be consistent even when the disfluency is filler words or repetitive tokens. Table 3 contains a sentence from the annotated data, which depicts this issue for repetition. In the German source sentence, the word *üblicherweise*, meaning 'customarily' is annotated as a disfluency, as it was the speaker's intention to change the utterance into the next word *traditionell*, which means 'traditionally'.

Table 3: *Difficulty in detecting repetitions*

Source	Die Kommunikation zwischen Mensch und Maschine, die wir so **üblicherweise** traditionell immer sehen, ist die...
Engl. gloss	The communication between man and machine, which we **customarily** traditionally always see, is the...

Discourse markers can be hard to capture, as they occasionally convey meanings in a sentence. In the same way as it is with English discourse markers such as "I mean", "actually", and "like", for example, German discourse markers, as shown in Table 4, can sometimes be used as a discourse marker and sometimes as normal tokens. In this table it is shown that a German word *nun* means 'now' as shown in the upper part, but occasionally is used as a discourse marker like in the lower part and does not need to be translated. In the lower row, the word *nun* appears with another discourse marker *ja*, which can also mean 'yes' in English, depending on the context.

Table 4: *Difficulty in detecting discourse markers*

Source	Sie sehen hier unseren Simultanübersetzer, der **nun** meinen Vortrag transkribiert.
Reference	Here you see our simultaneous translator, which **now** transcribes my presentation.
Source	An einer Universität haben wir **ja nun** viele Vorlesungen.
Reference	In a university, we have many lectures.

These examples suggest that disfluency detection requires an analysis of syntactics as well as semantics. Detecting restarted fragments especially requires semantic labeling, as in some cases the restarted new fragment does not contain the same content as the aborted utterances.

In this work we aim to analyze and improve machine translation performance by detecting and removing the disfluencies in a preprocessing step before translation. For this we adopt a conditional random field (CRF)-based approach, in which the characteristics of disfluencies can be modeled using various features. In order to consider the issues discussed previously, we devised features learned from word representations and phrase tables used for the MT process in addition to lexical and language model features. The MT performance of CRF-detected output is evaluated and compared to the result of an oracle experiment, where the test data without all annotated disfluencies is translated.

This paper is organized as follows. In Section 2, a brief overview of past research on speech disfluency detection is given. The annotated data used in this work is described in Section 3, followed by Section 4 which contains the CRF modeling technique with extended features from word representation and phrase table information. Section 5 describes our experiment setups and their results along with an analysis. Finally, Section 6 concludes our discussions.

2. Related Work

In previous work, the disfluency detection problem has been addressed using a noisy channel approach [4]. In this work it is assumed that fluent text, free of any disfluencies passed a noisy channel which adds disfluencies to the clean string. The authors use language model scores and five different models to retrieve the string, where the two factors are controlled by weight. An in-depth analysis on disfluency removal using this system and its effect are provided in [5]. They find that for the given news test set, an 8% improvement in BLEU [6] is achieved when the disfluencies are removed.

In another noisy channel approach [7], the disfluency detection problem is reformulated as a phrase-level statistical machine translation problem. Trained on 142K words of data, the translation system translates noisy tokens with disfluencies into clean tokens. The clean data contains new tags of classes such as repair, repeat, and filled pauses. Using this translation model based technique, they achieve their highest F-score of 97.6 for filled pauses and lowest F-score of 40.1 for repairs.

The noisy channel approach is combined with a tree-adjoining grammar to model speech repairs in [1]. A syntactic parser is used for building a language model to improve the accuracy of repair detection. Same or similar words in roughly the same order, defined *rough copy*, are modeled using crossed word dependencies. Trained on the annotated Switchboard corpus, they achieve an F-score up to 79.7.

The automatic annotation generated in [1] is one of the features used for modeling disfluencies in [2], where they train a CRF model to detect speech disfluencies. In addition to the automatic identification by [1], they use lexical, language model, and parser information as features. The CRF model is trained, optimized and tested on around 150K words of annotated data, where disfluencies are to be classified into

three different classes. Following this work, the authors offer an insightful analysis on syntactics and semantics of manually reconstructed spontaneous speech [8].

Though most of the progress has been focused on enhancing the performance of speech recognition via disfluency detection, authors of the work [3] employ disfluency detection to achieve improved machine translation. They train three different systems. The first system combines hidden-event language models and knowledge-based rules. The second system is a CRF model, which combines lexical features and shallow syntactic features. The final system is a rule-based filler-detecting system. Five classes are used in this task. The test sets for testing MT performance are generated by manually pulling out sentences with disfluencies from all sentences available. Thus, only the sentences containing disfluencies are selected and evaluated. There are two test sets built in this way, which are 339 sentences and 242 sentences out of 1,134 sentences and 937 sentences respectively. Absolute improvements of 0.8 and 0.7 BLEU points are gained on the two selected test sets.

There are several notable differences between our disfluency detection system and previous work. Unlike [2], we deploy extended features from neural networks and a phrase table in order to capture more semantic aspects. Furthermore, in our work the CRF detection result is further processed and evaluated in an MT system. In the work in [3], three systems are combined to detect disfluencies and evaluated in an MT system. Contrary to their systems, we did not deploy any rule-based detection. Moreover, in our work the CRF-based disfluency detection is extended further using semantic features. Finally, in contrast to using only the affected 28% portion of their test data to evaluate the MT performance, we use all our available data for evaluation, including unaffected, originally clean sentences. This aims at evaluating the performance in a more fair condition.

3. Data

For training and testing our CRF model for disfluency detection, we use in-house German lecture data from different speakers, which is transcribed, annotated, and translated into English.

Disfluencies are annotated manually on a word or phrase level. There are subcategories of annotation such as filler words, repetitions, deletions, partial words, and so on. These subcategories are very fine-grained, so we later re-classify them for the CRF tagging task according to our aims. Inspired by the classes defined in previous works [1, 2], we classified these annotations into three categories; `filler`, `(rough)copy`, and `non-copy`.

The class `filler` includes simple disfluencies such as *uhm*, *uh*, *like*, *you know* in English. If source words are discourse words or do not necessarily convey meaning and are not required for correct grammar, they are also classified as filler words. Words or phrases are grouped into `(rough)copy` when the same or similar tokens reoccur, as shown in Table 1 and 3 with bold letters. Words are tagged as `non-copy` when the speaker changes their mind about how or what to say, as shown in Table 2 with bold letters. Contrary to previous work [2], extreme cases of `non-copy`, in which the restarted fragments are considered to have new contexts after aborted utterances, are not excluded from the modeling target but are also taken into account.

Compared to other works on English, we have a considerably lower amount of annotated data in German. We gathered 61K manually-annotated words from lecture data, with roughly 9% marked as disfluencies. Detailed statistics of the annotated data is given in Table 5.

Table 5: *Data statistics on classes of the annotation*

	Tokens	Percentage in the corpus
`Filler`	3,304	5.35%
`(rough)Copy`	1,518	2.46%
`Non-copy`	620	1.00%
Non-disfluency	56,264	91.18%

In order to make use of all annotated data and to enable cross validation, we divided the 61K words of annotated data as well as its translation in English into three parts, such that each part has around 20K words in the German source. For testing one corpus part out of three, the other two parts, which are around 40K words, are used as training data for the CRF model.

4. Disfluency Detection using CRF

Introduced by [9], CRF is a framework dedicated to labeling sequence data. A CRF models a hidden label sequence given the observed sequence. CRFs have been applied extensively in diverse tasks of NLP, such as sentence segmentation [10], POS tagging [9] and shallow parsing [11] due to its advantages of representing long-range dependencies in the observations.

In this work we use the linear chain CRF modeling technique to detect speech disfluencies. By using bigram features we can model first-order dependencies between words with a disfluency. We used the GRMM package [12] implementation of the CRF model. The CRF model was trained using L-BFGS, with the default parameters of the toolkit.

4.1. Features

In this work we utilize lexical, language model, word representation, and phrase table information features. Word representation and phrase table information features are devised in order to capture more syntactic and semantic characteristics of speech disfluencies. They are described in detail later on.

Our lexical and language model features are based on the ones described in [2]. We extend the language model features on words and POS tags up to 4-grams. Parser information and JC-04 Edit results as shown in [1] are not available in

Table 6: *Sample features on the lexical level*

Source	Da	gibt	es	da	gab	es	in	uh	gab	es	nur	eins	.
Engl. gloss.		There is			there was		in	uh	there was		only	one	.
Word	Da	gibt	es	da	gab	es	in	uh	gab	es	nur	eins	.
POS	ADV	VVFIN	PPER	ADV	VVFIN	PPER	APPR	ITJ	VVFIN	PPER	ADV	PIS	$.
Word-Dist	3	365	3	47	4	4	259	9	218	821	115	933	27
POS-Dist	3	3	3	7	4	4	12	9	6	80	3	21	27
Word-Patt	0	0	0	0	1	1	0	0	0	0	0	0	0
POS-Patt	1	1	1	0	1	1	0	0	0	0	0	0	0
Annotation	–	RC	RC	RC	RC	RC	RC	FL	–	–	–	–	–

German, and therefore not used in this work. Furthermore, we add two new pattern features on the lexical level.

In Table 6, several selected features are shown for the rough repetition sentence from Table 1. The 'Word/POS-Dist' feature means the distance of a token to its next appearance. Therefore, a low 'Word/POS-Dist' number indicates that this token occurs again shortly thereafter. If two or more neighboring tokens have the same 'Word/POS-Dist', the 'Word/POS-Patt' feature of the corresponding tokens is set to 1. For example, the first three tokens have the same 'POS-Dist' number, therefore their 'POS-Patt' has a value of 1. This feature enables us to efficiently detect such blocks of repetition, where the same or roughly the same words are repeated. We use a 1 of k encoding for features. Since binary features are supported better by the toolkit, we quantize the numeric features. The POS tags are automatically generated using [13].

With the mentioned features, we can find syntactic clues for disfluency detection. For example, POS tokens and their patterns can help to figure out repetitive `(rough)copy` occurrences. However, as discussed earlier, in the annotated data we observe that in many cases it is required to include a semantic level of information as well. In addition to the mentioned features, we devised a new strategy of including word embedding features derived from a recurrent neural network (RNN) and phrase table information.

4.2. Word Representation using RNN

Word representations have gained a great deal of attention for various NLP tasks. Especially word representation using RNNs is proven to be able to capture meaningful syntactic and semantic regularities efficiently [14]. RNNs are similar to multilayer perceptrons, but an RNN has a backwards directed loop, where the output of hidden layers becomes additional input. This allows the network to effectively capture longer history compared to other feed-forward-based n-gram models.

Word embedding is a distributed word representation, where words are represented as multi-dimensional vectors. The word vectors syntactically and semantically relating to each other will be close to each other in that representation space. Thus, words within certain semantic and syntactic relations have similar vector values. Conventionally, word embeddings of a textual corpus are obtained using certain types of neural networks.

In the hope that word representation can offer insights on semantics and syntactis, in this paper we use word embedding features learned from an RNN for the CRF model. We use RNNLM [15] with 100 dimensions for word representations. In order to ensure an appropriate coverage of the representation, we use the preprocessed training data of the MT system, which contains various domains such as news and lectures. This data consists of 462 million tokens with 150K unique tokens.

4.2.1. Word Projection and Cosine Distance

Figure 1 depicts the 2-dimensional word projection from the real-valued 100-dimensional vectors representations using the RNN, we can observe word clusters being formed. This visualization is obtained using t-Distributed Stochastic Neighbor Embedding [16]. Due to memory consumption, only the most frequent 10K words are projected.

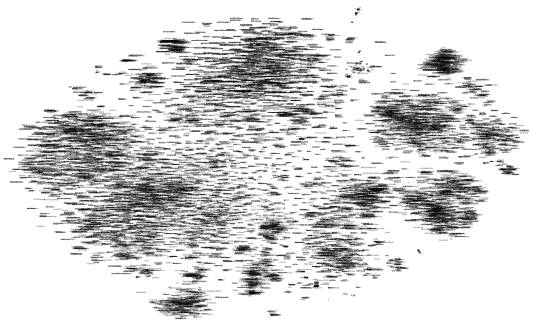


Figure 1: *Word projection of training data, with word representation obtained with an RNN*

Analyzing the details of this projection, we observe that words with the same syntactic role are projected closely to each other. For example, possessive cases corresponding to 'my', 'his', and 'our' in English are projected closely to each other. This is consistent for other grammatical components of a sentence, such as personal pronouns or relative pronouns. We observe clusters for dates, months and times.

The projection seems to convey semantic relations to

some extent. When it comes to adjectives, they are projected according to their stem and occasionally meanings. Verbs are clustered with other verbs with the same tense or stem.

In order to compare the closeness of words numerically, we calculate their cosine similarity.

Table 7: *Cosine similarity of words in word representations*

Word in German	Meaning in English	Cosine Distance
schnell	fast, quick	1
rasch	quick, rapid	0.8394
bald	soon, shortly	0.6245
effektiv	effective	0.6092
zügig	efficient, speedy	0.6088
wahrscheinlich	probable	1
vermutlich	probably	0.9066
möglicherweise	maybe, possibly	0.8938
sicherlich	certainly	0.8937
vielleicht	maybe, possibly	0.8827

Table 7 depicts a couple of examples. For each bold-lettered word, the four words with the highest cosine similarity are presented. Evidently, these four words are sharing a high semantic closeness with each given word, which will provide a quality feature for the task of disfluency detection. From this analysis, we conclude that RNNs can offer syntactic and semantic clues for disfluency detection.

4.2.2. Word Clustering

In order to use the word representation vectors as features in the CRF model more efficiently, we cluster the word representations with the k-means algorithm. From preliminary experiments, the number of clusters k is chosen to be 100.

Therefore every word of the RNN training data falls into the 100 clusters. For every word in the test data, it is checked whether this word has been observed in the word representations. If it has been observed, the word is assigned with the corresponding cluster code as a binary feature. If it has not been observed, the cluster code 0 is assigned. Also, the distance to the next identical cluster code and the repetitive pattern of it are also used as CRF model features, as shown in Table 6 for word and POS tokens.

4.3. Phrase Table Information

One of the common effects of disfluencies on the MT process is that often the translation contains repetitive words or phrases. When identical tokens in the source sentence are the reason for this, the original source sentence can be corrected using lexical features. However, often we observe other cases where two words, which are different on the lexical level, generate two identical translated words. Table 8 depicts one example for this from our data.

In this example, the German word *jetzt* (Engl. gloss. 'now') is annotated as a disfluency, followed by a word *inzwischen* (Engl. gloss. 'meantime', 'now'). Translating this source sentence as it is generates the translation containing two identical tokens in a row in English. We expect to solve this problem by examining the meaning of the source words in a phrase table. Thus, the target words for given source words in a phrase table are examined.

An advantage from using phrase table information is that we can detect semantic closeness of words or phrases in a source sentence independent from their syntactic roles. As shown in Table 7, word representation tends to group those words together which are syntactically and semantically closely related. However, using the phrase table information, words which are only semantically related, but not necessarily syntactically related, can also be grouped together. Considering that many of the repetitions also have different POS tags in a sentence, this phrase table feature is expected to capture such disfluencies.

In order to derive this feature, we examine the bilingual language model [17] tokens in the phrase table. The bilingual language model tokens consist of target words and their aligned source words. Using this information, we count how often a given source word is aligned to a certain target word and list the three most frequently used target words. We compare the aligned target words of the current and the following word. If the same target word(s) appears in both lists, the current word is given a phrase table feature.

An equivalent feature is introduced for the phrase level. As an example, we can consider consecutive source words f_1, f_2, and f_3 in one phrase. This phrase is aligned to a target token e_1. If the next source token f_4 is also aligned to the target token e_1, the first three tokens, namely f_1, f_2, and f_3, are given the phrase level phrase table feature. The coverage of the phrase level feature can be expanded upto three consecutive words as one phrase on the source side. Thus, the source tokens f_1, f_2, and f_3 are examined as one phrase, and this can be also narrowed down to f_1 and f_2 only. The target token(s) aligned to the source phrase, consists of upto f_1, f_2, and f_3, is compared to the target token(s) aligned to the potential repetitive phrase, which can consist of also upto next three tokens f_4, f_5, and f_6. The German source words with split compounds are also considered in this way.

In our phrase table the word *inzwischen* in Table 8 is aligned to 'now' most frequently, followed by 'meantime' and 'meanwhile'. The most frequently appeared translation for the next appearing word *jetzt* is 'now', followed by 'currently', and 'just'. Thus, by using the phrase table features, it will be indicated that the first word *jetzt* is aligned to a same target word with its next appearing word.

5. Experiments

5.1. System Description

In this section we introduce the SMT system used in our experiments. The translation system is trained on 1.76 million sentences of German-English parallel data including the European Parliament data and the News Commentary corpus.

Table 8: *Necessity of using phrase table information for disfluency detection*

Source	Diese Vorlesungen sind natürlich **jetzt inzwischen** alle abgespeichert, die liegen auf unserem Server.
Engl. gloss	These lectures are of course **now meantime** all stored, they lie on our server.
MT output	This lecture series are, of course, **now now** all stored, which lie on our server.
Reference	These lectures have of course all been saved in the meantime, they are on our server.

We also use the parallel TED data[1] as in-domain data to adapt our models to the lecture domain. Preprocessing which consists of text normalization, tokenization, and smartcasing is applied before the training. For the German side, compound splitting and conversion of words written according to the old spelling conventions into the new form of spelling are applied additionally.

As development data, manual transcripts of lecture data collected internally at our university are used. The talks are 14K parallel sentences from university classes and events.

In order to build the phrase table, we use the Moses package [18]. Using the SRILM Toolkit [19], a 4-gram language model is trained on 462 million words from the English side of the data. A bilingual language model [17] is used to extend source word context. In order to address the different word orders between German and English, the POS-based reordering model as described in [20] is applied. This is further extended as described in [21] to cover long-range reorderings. We use Minimum Error Rate Training (MERT) [22] for the optimization in the in-house phrase-based decoder [23].

5.2. Results

To investigate the impact of disfluencies in speech translation quality, we conduct four experiments.

In the first experiment, the whole data, including annotated disfluencies, is passed through our statistical machine translation (SMT) system.

For the second experiment, we remove the obvious filler words *uh* and *uhm* manually in order to study the impact of the filler words which can be captured systematically. Although there are a great number of other filler words, many of these filler words are not removed in this experiment, since they are not always disfluencies.

In the third experiment, we use the output from the CRF model trained with features from word representations and phrase table information, which will be noted as CRF-Extended. We also translate the output from the CRF model trained without any word representation and phrase table features. This will be denoted as CRF-Baseline. If the CRF models detect a token as either of the three classes, `filler`, `(rough)copy`, or `non-copy`, the word token is assumed to be a disfluency and is removed. The three classes are trained in the same model together. As mentioned previously, training and testing the CRF model is done with three-fold cross-validation. Thus, both of the CRF models are trained on around 40K annotated words, and tested on around 20K annotated words. The performance is evaluated on the joined three sub-test sets.

In the last experiment, all disfluency-annotated words are removed manually. As all annotation marks are generated manually, this experiment shows as an orcale experiment the maximum possible improvement we could achieve.

All four experiments are conducted on manually transcribed texts, in order to disambiguate the effects from errors of an ASR system. The experiments considers all available data, which is 61K words, or 3K sentences.

Table 9 depicts the results of our experiments. The scores are reported as case-sensitive BLEU scores, including punctuation marks.

Table 9: *Influence of disfluency in speech translation*

System	BLEU
Baseline	19.98
+ no *uh*	21.28
CRF-Extended	21.94
Oracle	23.14

The result of the first experiment is presented as the Baseline system, where all disfluencies are kept in the source text. When we remove all *uh*s and *uhm*s in the source text manually, we gain 1.3 BLEU points.

Apart from this, we use the output of the CRF-Extended as an input to our machine translation system. Words tagged as disfluencies are all removed. The translation score using the CRF-Extended is almost 2 BLEU points better than translating the text with all disfluencies. Compared to the second experiment where we remove *uh* and *uhm*, the performance is improved by around 0.7 BLEU points. As the BLEU score does not show a significant difference between the CRF-Extended and CRF-Baseline, here only the CRF-Extended score is shown. An in-depth analysis of the impact of the two systems will be given in the following chapter.

5.3. Analysis

The detection results for all models are given in Table 10. In total, there are 5,432 speech disfluencies annotated by human annotators, and among them, 3,012 speech disfluencies are detected by the CRF-Extended.

Compared to the case where the obvious filler words are removed, 1,025 more speech disfluencies are detected and removed. Compared to the CRF-Baseline, where the features obtained from the word representations and phrase table information are not used, 103 more disfluencies are detected

[1] `http://www.ted.com`

Table 10: *Results of disfluency detection in tokens*

System	Correct	Wrong
Baseline	0	0
+ no *uh*	1,987	0
CRF-Baseline	2,909	489
CRF-Extended	3,012	552
Oracle	5,432	0

using the CRF-Extended, while also a higher number of tokens are falsely detected.

In order to analyze the difference between the translations produced by CRF-Baseline and CRF-Extended, we score the test set sentence by sentence and rank them according to the difference in BLEU scores. Differences appear in 223 sentences.

One notable difference is that the CRF-Extended system detects a higher number of repetitions. Table 11 shows a sentence from the test set, where a longer phrase of repetition is captured using CRF-Extended. Words which represent a disfluency are marked in bold letters. Both systems can catch the obvious filler word *uh* and the simple repetition *als als*. In addition to this detection, the CRF-Extended system captures the whole disfluency region, in spite of the considerably complicated sentence structure and repetitive patterns. In this sentence the repeated words appear with varying frequencies and with a different distance to the next identical token. In order to detect such disfluencies, the correct phrase boundary needs to be recognized. As a result of this detection, the MT output using the CRF-Extended system is much more fluent than the one using the CRF-Baseline system.

Table 12 shows a sentence from the test set, where the CRF-Extended system does not perform better than the CRF-Baseline system for the given reference. The only disfluency shown in the original sentence *der*, marked with bold letters, is removed using both techniques. The CRF-Extended system additionally detects *einen Umschwung* as a disfluency. However, this deletion harms neither the structure nor meaning of the sentence, as *einen Umschwung* means 'a turnaround', or 'a change', which conveys practically the same meaning as the next following tokens.

It is an interesting point that using the semantic features we could detect that *einen Umschwung* is semantically closely related with *eine veränderte*, despite their distance in tokens and different syntactic roles in the sentence. This is an example that even though the CRF-Extended output does not match the human-generated annotation in this case, the CRF-Extended still provides a good criteria to detect semantically related words.

The CRF-Extended system also performs better with regard to distinguishing between discourse markers and the normal usages of the words. 59% of difference in correctly classified disfluencies between the CRF-Baseline and CRF-Extended stems from filler words. The rest is achieved from detecting a higher number of correct repetitions.

6. Conclusions

In this paper, we presented a CRF-based disfluency detection technique with extended features from word representations and a phrase table. These features are designed to capture deeper semantic aspects of the tokens. Using the predicted results from the CRF model, we gain around 2 BLEU points on manual transcripts of lectures. From the detailed analysis, we show that usage of the extended features provides a good means to detect semantically related disfluencies. The oracle experiment suggests that the machine translation of spontaneous speech can be improved significantly by detecting more disfluencies correctly.

Table 11: *Syntactically complicated, long phrase with a disfluency captured using CRF-Extended*

Source	Man kann das natürlich sowohl **als Links- als auch als** als Links- als auch als Rechtshänder **uh** verwenden.
Engl. gloss	You can this of course both **as left- as also as** as left- as also as right-handed **uh** use.
CRF-Baseline	Man kann das natürlich sowohl **als Links- als auch** als Links- als auch als Rechtshänder verwenden.
MT output	You can use this, of course, both as a left- as well as on the left- as well as a right-handed.
CRF-Extended	Man kann das natürlich sowohl als Links- als auch als Rechtshänder verwenden.
MT output	You can use this, of course, both as a left- as well as a right-handed.
Reference	You can of course use this as left- as well as also as a right-handed person.

Table 12: *Semantically related words detected using CRF-Extended*

Source	Die Ausrufung des totalen Kriegs markierte eigentlich *einen Umschwung*, **der** *eine veränderte* Form der Politik.
Engl. gloss	The proclamation of total war marked actually *a turnaround*, **of** *a change* form of politics.
CRF-Baseline	Die Ausrufung des totalen Kriegs markierte eigentlich *einen Umschwung, eine veränderte* Form der Politik.
MT output	The proclamation of the total war was collared actually *a turnaround, a changed* form of politics.
CRF-Extended	Die Ausrufung des totalen Kriegs markierte eigentlich *eine veränderte* Form der Politik.
MT output	The proclamation of the total war was collared actually *a changed* form of politics.
Reference	The call for total war in fact marked *a turnaround*, and *a changed* form of politics.

In future work, we would like to pursue the development of disfluency detection systems which take prosodic features into account in order to apply them to automatic speech recognition output. Furthermore, integrating the disfluency detection system tightly into machine translation systems could improve the performance even more.

7. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

8. References

[1] M. Johnson and E. Charniak, "A TAG-based Noisy Channel Model of Speech Repairs," in *ACL*, 2004.

[2] E. Fitzgerald, K. Hall, and F. Jelinek, "Reconstructing False Start Errors in Spontaneous Speech Text," in *EACL*, Athens, Greece, 2009.

[3] W. Wang, G. Tur, J. Zheng, and N. F. Ayan, "Automatic Disfluency Removal for Improving Spoken Language Translation," in *ICASSP*, 2010.

[4] M. Honal and T. Schultz, "Correction of Disfluencies in Spontaneous Speech using a Noisy-Channel Approach," in *Eurospeech*, Geneva, 2003.

[5] S. Rao, I. Lane, and T. Schultz, "Improving Spoken Language Translation by Automatic Disfluecy Removal: Evidence from Conversational Speech Transcripts," in *Machine Translation Summit XI*, 2007.

[6] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation." IBM Research Division, T. J. Watson Research Center, Tech. Rep. RC22176 (W0109-022), 2002.

[7] S. Maskey, B. Zhou, and Y. Gao, "A Phrase-Level Machine Translation Approach for Disfluency Detection using Weighted Finite State Tranducers," in *Interspeech*, 2006.

[8] E. Fitzgerald, F. Jelinek, and R. Frank, "What Lies Beneath: Semantic and Syntactic Analysis of Manually Reconstructed Spontaneous Speech," in *ACL*, 2009.

[9] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilitic Models for Segmenting and Labeling Sequence Data," in *ICML*, Massachusetts, USA, 2001.

[10] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Using Conditional Random Fields for Sentence Boundary Detection in Speech," in *ACL*, Ann Arbor, MI, 2005.

[11] F. Sha and F. Pereira, "Shallow Parsing with Conditional Random Fields," in *HLT/NAACL*, 2003.

[12] C. Sutton, "GRMM: A Graphical Models Toolkit," 2006. [Online]. Available: http://mallet.cs.umass.edu

[13] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

[14] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," in *NAACL-HLT*, 2013.

[15] T. Mikolov, M. Karafiat, J. Cernocky, and S. Khudanpur, "Recurrent Neural Network based Language Model," in *Interspeech*, 2010.

[16] L. van der Maaten and G. Hinten, "Visualizing High-Dimensional Data using t-SNE," in *Journal of Machine Learning Research 9*, 2008.

[17] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *WMT*, Edinburgh, UK, 2011.

[18] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *ACL 2007, Demonstration Session*, Prague, Czech Republic, June 2007.

[19] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *Proc. of ICSLP*, Denver, Colorado, USA, 2002.

[20] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *TMI*, Skövde, Sweden, 2007.

[21] J. Niehues and M. Kolss, "A POS-Based Model for Long-Range Reorderings in SMT," in *WMT*, Athens, Greece, 2009.

[22] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *WPT-05*, Ann Arbor, MI, 2005.

[23] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *Int. Conf. on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

Maximum Entropy Language Modeling for Russian ASR

Evgeniy Shin, Sebastian Stüker, Kevin Kilgour,
Christian Fügen, Alex Waibel

International Center for Advanced Communication Technology
Institute for Anthropomatics, Karlsruhe Institute of Technology
Karlsruhe, Germany

Abstract

Russian is a challenging language for automatic speech recognition systems due to its rich morphology. This rich morphology stems from Russian's highly inflectional nature and the frequent use of pre- and suffixes. Also, Russian has a very free word order, changes in which are used to reflect connotations of the sentences. Dealing with these phenomena is rather difficult for traditional n-gram models. We therefore investigate in this paper the use of a maximum entropy language model for Russian whose features are specifically designed to deal with the inflections in Russian, as well as the loose word order. We combine this with a sub-word based language model in order to alleviate the problem of large vocabulary sizes necessary for dealing with highly inflecting languages. Applying the maximum entropy language model during re-scoring improves the word error rate of our recognition system by 1.2% absolute, while the use of the sub-word based language model reduces the vocabulary size from 120k to 40k and the OOV rate from 4.8% to 2.1%.

1. Introduction

The Russian language has some properties that make the creation of high performing *Large Vocabulary Continuous Speech Recognition* (LVCSR) quite challenging. Especially in language modeling there are two principal problems that need to be dealt with:

- *Morphology:* Russian is a highly inflecting language. E.g., Russian nouns can be declined according to six cases, two numbers (singular and plural) and three grammatical genders (male, female and neutral). Adjectives need to declined in accordance with the subject that they belong to; verbs can be conjugated according to three persons, two numbers and two tenses. Prefixes and suffixes are frequently used to produce a multitude of derivatives of basic words.

- *Word Order:* The word order in Russian is rather free. Different word orders for the same sentence are used to convey different connotations.

The rich morphology of Russian leads to the need for large vocabularies. And even with rather large vocabularies ASR systems suffer from relatively high *out of vocabulary* (OOV) rates [1, 2].

Also, the combination of loose word order and rich morphology leads to very high perplexities for standard n-gram language models, especially when trained estimated on moderate amounts of training data [1, 3]. Larger vocabularies generally lead to higher n-gram language model perplexities. The same is true for the loose word order, as n-gram language models compose the sentence language model probability from the probabilities of word sequences of fixed order and short length.

In order to deal with the problem of high OOV rates that arise from the rich morphology of a language, the use of sub-word based search vocabularies is a common technique and has been successfully used in a multitude of languages (see Section 2). However, their impact on the problems of the high perplexities of the language model are only limited, especially for Russian with respect to its many endings arising from the grammatical inflections, but also with respect to its many prefixes and suffixes that can be combined with a myriad of words.

In order to alleviate this problem we propose the application of maximum entropy language models to Russian. In this paper we present an implementation of such a maximum entropy language model that deals specifically with the phenomena that make n-gram language models perform badly for Russian. We combine the maximum entropy model with our implementation of a sub-word based vocabulary and evaluate both approaches on a large vocabulary continuous speech recognition task in the tourist domain.

The rest of the paper is structured as follows. In Section 2 we give an overview of related work in both areas – sub-word based language modeling and maximum entropy language models. Section 3 then introduces our approach to sub-word based language modeling for Russian, while Section 4 describes our design of an entropy based language model that deals specifically with Russian morphology. In Section 5 we report on the improvements in word error rate that we achieved with the approaches described in this paper.

2. Related Work

2.1. Sub-Word Based Language Models

Sub-word based language models have been reported to be successful for highly inflecting languages such as Russian[4, 1], Czech[5], Finnish[6], Turkish[7], Slovenian[8], Arabic[9, 10].

In [9] *SyntaxNN*, a neural network language model using syntactic and morphological features, and *DLM*, a discriminative language model trained using the Minimum Bayes Risk (MBR) criterion, and unigram, bigram, and trigram morphs features were applied to Arabic.

To incorporate syntactical and morphological knowledge of Arabic to language modeling [10] utilized a *Factored Language Modeling* toolkit[11]. The use of word lexeme and morpheme features led to a reduction in WER of 2% relative.

A particle (similar to sub-word) based n-gram model in combination with a word based model applied to Russian was shown to give a reduction of perplexity of up to 7.5% [4]. For this, data-driven techniques were applied that determine particle units and word decompositions automatically.

A random-forest language model for Russian[4] using word stems among other morphological features achieved a WER improvement of 3.4% relative over a trigram model.

[12] explored the use of sub-word based language models for Finnish, Estonian, Turkish and Egyptian Colloquial Arabic. They performed word decomposition in an unsupervised, data-driven way using *Morfessor*. They showed that the morph models performed fairly well on OOVs without compromising the recognition accuracy of in-vocabulary words.

An application of sub-word based language model to Czech is studied in [5]. A sub-word based language model which includes different models for different sub-word units, such as stems and endings, reduces the WER by about 7% absolute. They applied their language model in n-best list re-scoring.

An interesting idea is proposed in [7]. Here, Turkish was modeled with so called FlexGrams, which allow skipping several parents and use later grams in the history to estimate a probability of the current word. They experimented with words split into their stem and suffix forms, and defined stem-suffix FlexGrams, where one set of offsets is applied to stems and another to suffixes.

2.2. Maximum Entropy Language Models

The maximum entropy approach was introduced to language modeling more than 10 years ago[13, 14, 15]. And it is being used today the state-of-the-art language models such as ModelM[16].

ModelM[16] is an exponential class-based n-gram language model. The word n-gram and word class features are incorporated into the language model within an exponential modeling framework. The model with enhanced word classing[17] achives a total gain of up to 3.0% absolute over a Katz-smoothed trigram model[17]. Experiments were done on the Wall Street Journal corpus.

Maximum Entropy models are also being successfully used for machine translation systems, e.g. [18, 19]

In [19] it was shown that the use of discriminative word lexica (DWL) can improve the translation quality significantly. For every target word, they trained a maximum entropy model to determine whether this target word should be in the translated sentence or not. As features for their classifier they used one feature per source word.

3. Sub-Word Based Search Vocabulary and Language Model

The goal of sub-word based search vocabularies and language models is to reduce the OOV rate of an ASR system by decomposing whole words into smaller units. Normally, the distinct number of these sub-word units is significantly smaller than the number of words that they form. So, with constant vocabulary size, the OOV rate of the recognition system is drastically reduced.

In order to work, the following steps need to be taken:

- *Decomposition:* The original words need to be decomposed into smaller units. The units need to show some sort of consistency, so that their total number is clearly smaller than that of the words that they were derived from. Depending on the language one can decide to either decompose all words in the search vocabulary, or only a certain sub-set, e.g., those occurring relatively infrequently, while the frequent words are being kept intact. Word decomposition is usually done for the language model training material and then a new vocabulary is derived.

- *Pronunciation Generation:* For the generated sub-word units pronunciations need to be added to the system's dictionary. Since in general the mapping between the writing of a word and its pronunciation, i.e. phoneme sequence, is not given or easily derivable, deducting the pronunciation of the sub-word units from the pronunciation of the original words is often not straight-forward or even impossible. Often grapheme based pronunciation dictionaries can offer a solution here.

- *Language Model Training:* Based on the new vocabulary composed of the sub-word units, and potentially mixed with whole words, a new language model needs to be trained that is then used for recognition.

- *Word Reconstruction:* After decoding, the recognized sub-words need to be recombined in order to obtain a valid word sequence.

3.1. Word Decomposition and Merging

For word decomposition we used a *Snowball* [20] based stemmer. Snowball is a small string processing language designed for creating stemming algorithms. A stemmer for Russian is distributed with the package. The stemmer is not a tool for morpheme analysis, but a word stem derivation tool. Therefore, the output of this tool needs to be processed to split up words into subunits. For a given word the stemmer returns a stem. Endings can then be derived by comparing the original word string against that of the stem. For example the words in the phrase "необходимое условие" (necessary conditions) are decomposed into:

word		stem		ending
необходимое	→	необходим	→	ое
условие	→	услов	→	ие

Compound words that are joined via a hyphen, are first split before being put through the stemmer, as every sub part of a compound might have its own ending.

In order to simplify the merging of sub-words after decoding every word part after the first stem is marked as an ending. After decoding all endings after a stem are merged to the stem, until a new stem is encountered. For words that do not have an explicit ending, the null-ending was utilized for language modeling.

4. Maximum Entropy Language Modeling

In maximum entropy modeling the model is constrained by features. In language modeling these features must be extractable from the word sequence for which the probability needs to be calculated. The models are then trained according to the maximum conditional entropy criterion. Thereby a number of different training algorithms are available for finding the probability distribution with the maximum entropy, given the training data.

4.1. Features

For n-gram models the features used are the bigrams, trigrams, etc. that appear in the word sequence. For maximum entropy language models one can use additional features, such as part of speech (POS) tags, different grammatical categories or topic information. All these kinds of features can be represented by binary feature functions or indicator functions.

A bigram feature can for example be expressed by the following indicator function:

$$f_1(x,y) = \begin{cases} 1, & if \quad y = "day" \quad and \quad x = "nice" \\ 0, & otherwise \end{cases}$$

The function, feature respectively, f_1 returns 1 for the word y and its context x, if y and x form the bigram "nice day".

Using large amounts of training data we can estimate the probability distribution $p_e(x,y)$ where x and y can take on all possible words in the search vocabulary. Now, with the help of p_e, we can estimate a mean value of feature f_1:

$$\mu(f_1) = \sum_{x,y} p_e(x,y) f_1(x,y) = \sum_{x,y} relfreq(x,y) f_1(x,y) \tag{1}$$

If the training data is sufficiently large, the mean value represents the expected value of the real distribution:

$$\mathbb{E}(f_1) = \sum_{x,y} p(x,y) f_1(x,y) \tag{2}$$

Our language model p_m is requested to be unbiased with respect to f_1, i.e. to have the same expected value for the feature f_1:

$$\sum_{x,y} p_e(x,y) f_1(x,y) = \sum_{x,y} p_m(x,y) f_1(x,y), \tag{3}$$

where $p_m(x,y)$ is the distribution as given by the model.

However, we are interested in modeling $p(y|x)$ and not $p(x,y)$. Therefore the constraint equations for feature f_1 has to be:

$$\sum_{x,y} p_e(x,y) f_1(x,y) = \sum_{x,y} p_e(x) p_m(y|x) f_1(x,y), \tag{4}$$

For every feature that we define for the maximum likelihood model such a constraint function is defined and has to be obeyed by our model distribution p_m.

4.2. Maximization of conditional entropy

Depending on which features we select for our language model, not only one but a whole set of distributions that comply with the constraints exists. From these many possible distributions the best one needs to be selected. One approach comes from information theory and is based on the concept of conditional entropy:

$$H(Y|X) = -\sum_{\substack{x \in X, \\ y \in Y}} p(x,y) \log p(y|x) \tag{5}$$

The idea of maximum entropy modeling is to choose that model which maximizes the conditional entropy of labels y given an information x (e.g., word context):

$$p_{me} = \arg\max_{p_m} H(p_m) \tag{6}$$

In simple words this means that the model makes no further assumptions about the given features. With the help of Lagrange multipliers, which are used to solve this constrained optimization problem, it can be shown that the resulting probability distribution has the parametric form:

$$p_{me}(\lambda) = \frac{1}{Z(x)} \exp\left(\sum_i \lambda_i f_i(x,y)\right), \tag{7}$$

where $f_i(x,y)$ are binary feature functions. λ_i are weight factors—parameters of the model. $Z(x)$ is the normalization factor in order to ensure that result is indeed a probability distribution.

4.3. Training

A number of algorithms can be used for estimating the parameters of a maximum entropy model. There are both——special methods, such as *Generalized Iterative Scaling*[21], *Improved Iterative Scaling*[22], and general purpose optimization techniques, such as gradient ascent, conjugate gradient and quasi-Newton methods. [23] in its comparison of algorithms for maximum entropy parameter estimation states that the widely used iterative scaling algorithms perform quite poorly, and for all of the test problems, a limited memory variable metric algorithm outperformed the other choices.

Four our experiments we used *Limited-memory BFGS* a limited memory variation of the *Broyden–Fletcher–Goldfarb–Shanno* (BFGS) method [24, 25], which is an implementation of the variable metric method. For this we used the *CRF++* Toolkit[26].

5. Experimental Set-Up and Results

We evaluated our two approaches on Russian data that was recorded by Mobile Technologies in the domain of tourist and basic medical needs, as it can be found in mobile speech translation devices such as Jibbigo[1]. We compare our results to a baseline with a word based n-gram model, while we keep the acoustic model fixed.

5.1. Data Set

The acoustic model training data accounts for about 620 hours of broadcast news and broadcast conversations acquired within the *QUAERO*[27] project. Further, we used a data set of read speech mostly in touristic and medical speech domains, provided by *Mobile Technology GmbH*[28]. From this set of 63 hours we cut away 3 hours as test set, while the rest went into acoustic model training.

For training our language models we used a text corpus collected from the Internet, 156M tokens in size. The text was crawled from forums in the touristic and medical domain.

The word decomposition for the sub-word based as well as the maximum entropy language model was done with the Snowball stemming algorithm[20].

Table 1 gives an overview for the datasets used.

AM training	Broadcast news & radio	620 hours
AM training	Read speech	60 hours
LM training	Web forums	156M words
Testing	Read speech	3 hours

Table 1: Over view over the acoustic data used for testing and AM training

[1] http://www.jibbigo.com

5.2. Baseline System

We performed all experiments with the help of the Janus Recognition Toolkit featuring the IBIS single pass decoder [29]. For our HMM based acoustic model we used a context dependent quinphone setup with three states per phoneme, and a left-to-right topology without skip states. The 8,000 models of the HMM were trained using *incremental splitting of Gaussians* (MAS) training, followed by *optimal feature space* training and 2 iterations of Viterbi training. The models were further improved with boosted MMIE training [30].

For the baseline system we used a standard 4-gram language model which we trained with the help of the SRI LM toolkit [31]. The search vocabulary was taken from the 120k most frequent words from the LM training data. For both cases the dictionaries are grapheme based dictionaries which works quite well for Russian [3].

5.3. Sub-Word Based Experiments

The sub-word based system uses a sub-word search vocabulary and a sub-word based 4-gram model. For this we split the words in the language model training with our procedure described in Section 3. As vocabulary we selected the 40k most frequent sub-word units.

5.4. Re-Scoring with Word N-Gram Model

While sub-word based language modeling reduces the OOV rate, it introduces additional problems such as a loss in language model reach, and the fact that the sub-word units are acoustically more confusable. Therefore, in order to combine the advantages of a sub-word based and a word based LM we re-scored n-best lists that were generated with the sub-word based LM.

Re-scoring was done by interpolating the combined acoustic and LM model scores of the sub-word based system with the LM score from the word based 4-gram LM. Interpolation was done as a weighted sum of the scores in the log domain. We tested a series of interpolation weights from 0 to 10.

5.5. Re-Scoring with Maximum Entropy LM

Word endings in Russian depend on several grammatical features of the current word, such as gender, case, tens, and form a pattern for the utterance. At the same time recognizing the endings correctly is quite challenging, as they have little acoustic evidence and are difficult to model with a regular n-gram LM. So, we selected features for the maximum entropy model that help with discriminating the endings. The features consist of words and endings in their context. Here

is a small example:

s_{-5}	e_{-5}	как	~#
s_{-4}	e_{-4}	подчеркнул	~#
s_{-3}	e_{-3}	офицер	~#
s_{-2}	e_{-2}	полиц	~ии
s_{-1}	e_{-1}	жёстк	~ие
s_0	e_0	мер	~ы
s_1	e_1	не	~#
s_2	e_2	применя	~лись

Since applying the entropy language model during regular decoding is too computationally intensive, again we applied the language model during n-best list re-scoring. For calculating the LM score we used the three previous stems (s_{-3}, s_{-2}, s_{-1}), three previous endings (e_{-3}, e_{-2}, e_{-1}) and one successor stem (s_1) and ending (s_1) as features. The null ending is explicitly modeled with the ~# place-holder.

For training, the *CRF++* Toolkit[26] is utilized. As the training of the labels, endings in our case, within a single model was not possible due to main memory usage (more than 512GB RAM was needed), a similar approach as in [18] and [19] was applied. The idea is to train a separate model for every label. Every model evaluates then only two classes: the ending, which the models stands for versus all other endings.

In testing, all models, whose corresponding endings were present in the utterance, were applied. The resulting score is given by the sum of the scores from the single models.

Again we re-scored the n-best lists generated by the sub-word system by interpolating the language model score from the maximum entropy language model with the combined acoustic and LM scores from the sub-word system. As for the interpolation described above we tested a series of interpolation weights, this time in the range of 0 to 20.

5.6. Results

5.6.1. Baseline System and Sub-Word Based System

Table 2 shows results of the full-word baseline and the sub-word based system. It can be seen that in spite of the fact that the OOV rate of the full-word system (4.8%) is higher than that of the sub-word system (2.1%), the latter performs slightly worse. Two of the reasons for that could be the higher acoustic confusability between the shorter sub-words and the shorter context of the sub-word based n-gram language model. The OOV rate of the sub-word based system is quite high but still half of that of the full-word system. The reason for that could be the difference in vocabulary size (40k vs. 120k).

	WER	OOV	vocabulary size
baseline	25.7%	4.8%	120k
sub-words	25.9%	2.1%	40k

Table 2: Word error rates, OOV rates and vocabulary sizes of the word based baseline and the sub-word based system

5.6.2. Re-Scoring with Word Based LM and Maximum Entropy LM

Figure 1 shows the result of our experiments in re-scoring the n-best lists from the sub-word system with a series of interpolation weights. One can see that for re-scoring with the word based LM, when choosing the right interpolation weight, we can improve the WER of the sub-word based system by 0.4% absolute.

When re-scoring with the maximum entropy model we can improve the WER of the sub-word based model by up to 1.2% absolute. We can also see that the interpolation is rather insensitive to the interpolation weight Finally, we combined

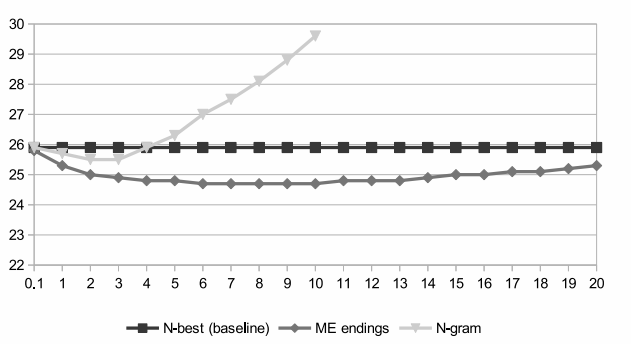


Figure 1: WER of re-scoring the n-best list of the sub-word system with the full word 4-gram model and with the maximum entropy model using different interpolation weights

both language models in the interpolation during re-scoring, taking the best interpolation weights from the individual re-scoring experiments. Table 3 shows the results of this combination. We can see that the improvements from the two language models sum up, i.e. their gains seem to be orthogonal to each other. In that way we can reduce the WER of the sub-word based based system by 1.6% absolute and that of our baseline system with the word based n-gram LM by 1.4% absolute.

Baseline	25.7%
Subwords	25.9%
+ Maximum entropy	24.7%
+ Word n-gram	24.3%

Table 3: Combined results of recognition and re-scoring systems

6. Conclusion

In this paper we investigated the use of a maximum entropy language model in order to deal with the highly inflectional nature of Russian and its loose word order. We designed the features of the language model specifically to target these problems. Applying the maximum entropy model during n-best list rescoring reduces the word error rate of our baseline

system by 1.2% absolute. In order to deal with the need for a large vocabulary for a Russian ASR system due to the many inflections possible in Russian, we implemented a sub-word based LM based on stemming. Using this language model reduces the vocabulary necessary during decoding from 120k to 40k and the OOV rate from 4.8% to 2.1%. By re-scoring the n-best lists of the sub-word based system with a combination of the maximum entropy language model and a word based 4-gram model, we can reduce the word error rate by another 0.2% absolute.

7. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No 287658 — 'Bridges Across the Language Divide' (EU-BRIDGE). Also, this work was in part realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation. 'Research Group 3-01' received financial support by the 'Concept for the Future' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative.

8. References

[1] E. Whittaker, "Statistical language modelling for automatic speech recognition of russian and english," *Daktaro disertacija, Cambridge University Engineering Department, Cambridge*, 2000.

[2] Y. Titov, K. Kilgour, S. Stüker, and A. Waibel, "The 2011 kit quaero speech-to-text system for the russian language," in *Proceedings of the 14th International Conference "Speech and Computer" (SPECOM'2011)*, September 2011.

[3] S. Stüker and T. Schultz, "A grapheme based speech recognition system for russian," in *Proceedings of the 9th International Conference "Speech And Computer" SPECOM'2004*. Saint-Petersburg, Russia: Anatolya, September 2004, pp. 297–303.

[4] I. Oparin, "Language models for automatic speech recognition of inflectional languages," Ph.D. dissertation, University of West Bohemia, 2008.

[5] P. Ircing, P. Krbec, J. Hajic, J. Psutka, S. Khudanpur, F. Jelinek, and W. Byrne, "On large vocabulary continuous speech recognition of highly inflectional language-czech," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[6] T. Hirsimäki, M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkönen, "Unlimited vocabulary speech recognition with morph language models applied to finnish," *Computer Speech & Language*, vol. 20, no. 4, pp. 515–541, 2006.

[7] D. Yuret and E. Biçici, "Modeling morphologically rich languages using split words and unstructured dependencies," in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Association for Computational Linguistics, 2009, pp. 345–348.

[8] T. Rotovnik, M. Maucec, and Z. Kacic, "Large vocabulary continuous speech recognition of an inflected language using stems and endings," *Speech communication*, vol. 49, no. 6, pp. 437–452, 2007.

[9] L. Mangu, H. Kuo, S. Chu, B. Kingsbury, G. Saon, H. Soltau, and F. Biadsy, "The ibm 2011 gale arabic speech transcription system," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 272–277.

[10] A. El-Desoky, R. Schlüter, and H. Ney, "A hybrid morphologically decomposed factored language models for arabic lvcsr," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 701–704.

[11] K. Kirchhoff, J. Bilmes, and K. Duh, "Factored language models tutorial," 2007.

[12] M. Creutz, T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraçlar, and A. Stolcke, "Morph-based speech recognition and modeling of out-of-vocabulary words across languages," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 5, no. 1, p. 3, 2007.

[13] A. Berger, V. Pietra, and S. Pietra, "A maximum entropy approach to natural language processing," *Computational linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[14] R. Rosenfield, "A maximum entropy approach to adaptive statistical language modeling," 1996.

[15] R. Rosenfeld, S. Chen, and X. Zhu, "Whole-sentence exponential language models: a vehicle for linguistic-statistical integration," *Computer Speech & Language*, vol. 15, no. 1, pp. 55–73, 2001.

[16] S. Chen, "Shrinking exponential language models," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009, pp. 468–476.

[17] S. Chen and S. Chu, "Enhanced word classing for model m," in *Proceedings of Interspeech*, 2010, pp. 1037–1040.

[18] M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel, "The kit english-french translation systems for iwslt 2011," in *Proceedings of the eight International Workshop on Spoken Language Translation (IWSLT)*, 2011.

[19] A. Mauser, S. Hasan, and H. Ney, "Extending statistical machine translation with discriminative and trigger-based lexicon models," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 210–218.

[20] M. Porter, "Snowball: A language for stemming algorithms," 2001.

[21] J. N. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *The annals of mathematical statistics*, vol. 43, no. 5, pp. 1470–1480, 1972.

[22] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Inducing features of random fields," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 4, pp. 380–393, 1997.

[23] R. Malouf *et al.*, "A comparison of algorithms for maximum entropy parameter estimation," in *Proceedings of the sixth conference on natural language learning (CoNLL-2002)*, 2002, pp. 49–55.

[24] M. Avriel, *Nonlinear programming: analysis and methods*. Courier Dover Publications, 2003.

[25] J. F. Bonnans, *Numerical optimization: theoretical and practical aspects: with 26 figures*. Springer-Verlag New York Incorporated, 2003.

[26] T. Kudo. (2005, Apr.) Crf++: Yet another crf tool kit. [Online]. Available: http://code.google.com/p/crfpp/

[27] (2008, Mar.) Quaero is a european research and development program. [Online]. Available: http://www.quaero.org/

[28] Mobile technologies gmbh. [Online]. Available: http://www.jibbigo.com/

[29] H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A One Pass-Decoder Based on Polymorphic Linguistic Context Assignment," in *ASRU*, Madonna di Campiglio Trento, Italy, December 2001.

[30] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted mmi for model and feature-space discriminative training," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4057–4060.

[31] A. Stolcke *et al.*, "Srilm-an extensible language modeling toolkit," in *Proceedings of the international conference on spoken language processing*, vol. 2, 2002, pp. 901–904.

Improved Speech-to-Text Translation with the Fisher and Callhome Spanish–English Speech Translation Corpus

Matt Post[*], *Gaurav Kumar*[†], *Adam Lopez*[*], *Damianos Karakos*[‡], *Chris Callison-Burch*[§], *Sanjeev Khudanpur*[†]

[*] Human Language Technology Center of Excellence, Johns Hopkins University
[†] Center for Language and Speech Processing, Johns Hopkins University
[‡] Raytheon BBN
[§] Computer and Information Science Department, University of Pennsylvania

Abstract

Research into the translation of the output of automatic speech recognition (ASR) systems is hindered by the dearth of datasets developed for that explicit purpose. For Spanish-English translation, in particular, most parallel data available exists only in vastly different domains and registers. In order to support research on cross-lingual speech applications, we introduce the Fisher and Callhome Spanish-English Speech Translation Corpus, supplementing existing LDC audio and transcripts with (a) ASR 1-best, lattice, and oracle output produced by the Kaldi recognition system and (b) English translations obtained on Amazon's Mechanical Turk. The result is a four-way parallel dataset of Spanish audio, transcriptions, ASR lattices, and English translations of approximately 38 hours of speech, with defined training, development, and held-out test sets.

We conduct baseline machine translation experiments using models trained on the provided training data, and validate the dataset by corroborating a number of known results in the field, including the utility of in-domain (information, conversational) training data, increased performance translating lattices (instead of recognizer 1-best output), and the relationship between word error rate and BLEU score.

1. Introduction

The fields of automatic speech recognition (ASR) and machine translation (MT) share many traits, including similar conceptual underpinnings, sustained interest and attention from researchers, remarkable progress over the past two decades, and resulting widespread popular use. They both also have a long way to go, with accuracies of speech-to-text transcription and text-to-text translation varying wildly across a number of dimensions. For speech, these variables determining success include properties of the channel, the identity of the speaker, and a host of factors that alter how an individual speaks (such as heartrate, stress, emotional state). Machine translation accuracy is affected by different factors, such as domain (e.g., newswire, medical, SMS, speech), register, and the typological differences between the languages.

Because these technologies are imperfect themselves, their inaccuracies tend to multiply when they are chained together in the task of speech translation. Cross-lingual speech applications are typically built by combining speech recognition and machine translation systems, each trained on disparate datasets [1, 2]. The recognizer makes mistakes, passing text to the MT system with vastly different statistical properties from the parallel datasets (usually newswire or government texts) used to train large-scale translation systems, which are then further corrupted with the MT system's own mistakes. Errors compound, and the results are often very poor.

There are many approaches to improving this speech-to-text pipeline. One is to gather training data that is closer to the test data, perhaps by paying professionals or using crowdsourcing techniques. The latter has been repeatedly demonstrated to be useful for collecting relevant training data for both speech and translation [3, 4, 5, 6], and in this paper we do the same for speech-to-text translation, assembling a four-way parallel dataset of audio, transcriptions, ASR output, and translations. The translations were produced inexpensively by non-professional translators using Amazon's popular crowdsourcing platform, Mechanical Turk (§2).

A second approach is to configure the ASR system to expose a portion of its search space by outputting more than just the single best output. Previous in speech-to-text translation have demonstrated success in translating ASR n-best lists [7] and confusion networks[1] [8], and lattices [9, 10]. In this paper, we apply similar techniques in the context of a machine translation, demonstrating consistent improvements over the single-best ASR translation in two different speech corpora.

The contributions of this paper are as follows:

- We extend two LDC Spanish speech sets with English translations and ASR recognizer output (in the form of lattices, ASR 1-best output, and lattice oracle paths) providing the community with a 3.8 million

[1] A confusion network, colloquially referred to as a *sausage*, is a restricted form of lattice in which all of a node's outgoing arcs go to the same head node.

word dataset for further research in Spanish-English speech-to-text translation.[2]

- We demonstrate improvements of up to 11.1 BLEU points in translating ASR output using this in-domain dataset as training data, compared to standard machine translation training sets (of twenty times the size) based on out-of-domain government and newswire text.

- We show further improvements in translation quality (1.2 absolute BLEU points) when translating the lattices instead of ASR 1-best output.

2. Collecting Translations

Here we describe the procedure used to obtain the translations, based on the current best practices for the collection of crowd-sourced translations.

The source data are the Fisher Spanish and Callhome Spanish datasets, comprising transcribed telephone conversations between (mostly native) Spanish speakers in a variety of dialects. The Fisher Spanish corpus[3] consists of 819 transcribed conversations on a variety of provided topics primarily between strangers, resulting in approximately 160 hours of speech aligned at the utterance level, with 1.5 million tokens. The Callhome Spanish corpus[4] comprises 120 transcripts of spontaneous conversations primarily between friends and family members, resulting in approximately 20 hours of speech aligned at the utterance level, with just over 200,000 words (tokens) of transcribed text. The combined dataset features a large variety of dialects, topics, and familiarity level between participants.

2.1. Crowdsourced Translations

We obtained translations using the popular crowdsourcing platform Amazon Mechanical Turk (MTurk), following a widespread trend in scientific data collection and annotation across a variety of fields [11, 12, 13, 14, 15, 3], and in particular the translation crowdsourcing work of [16].

We began by lightly preprocessing the transcripts, first to remove all non-linguistic markup in the transcriptions (such as annotations for laughter or background noise), and second to concatenate sequential utterances of a speaker during a single turn. Many utterances in the original transcript consisted only of single words or in some cases only markup, so this second step produced longer sentences for translation, enabling us to provide more context to translators and reduce cost. When the length of a combined utterance exceeded 25 words, it was split on the next utterance boundary.

We present sequences of twenty of these combined utterances (always from the same transcript) in each individual translation task — human intelligence tasks (HIT), in MTurk terminology. The utterances in each HIT were presented to each translator in the original order alongside the speaker name from the source transcript, thereby providing the translators with context for each utterance. HITs included the instructions taken from [16].

2.2. Quality Control Measures

MTurk provides only rudimentary tools for vetting workers for a specialized task like translation, so following established practice, we took steps to deter wholesale use of automated translation services by our translators.

- Utterances were presented as images rather than text; this prevented cutting and pasting into online translation services.[5]

- We obtained translations from Google Translate for the utterances before presenting them to workers. HITs which had a small edit distance from these translations were manually reviewed and rejected if they were too similar (in particular, if they contained many of the same errors).

- We also included four consecutive short sentences from the Europarl parallel corpus [17] in each HIT. HITs which had low overlap with the reference translations of these sentences were manually reviewed and rejected if they were of low quality.

We obtained four redundant translations of sixty randomly chosen conversations from the Fisher corpus. In total, 115 workers completed 2463 HITs, producing 46,324 utterance-level translations and a little less than half a million words.

2.3. Selection of Preferred Translators

We then extended a strategy devised by [16] to select high-quality translators from the first round of translations. We designed a second-pass HIT which was used to rate the above translators; those whose translations were consistently preferred were then invited to subsequent Spanish-English translation tasks.

For this voting task, monolingual English-speaking workers were presented with four different translations of an input sentence or utterance and asked to select the best one. As with the first HIT, users were presented with a sequence of twenty utterances from the same conversation, thereby providing local context for each decision. Each HIT was completed by three workers; in total, 193 workers completed 1676 assignments, yielding 31,626 sentence-level comparisons between 4 alternative translations.

From this data, we qualified 28 translators out of the initial 115. This set of translators produced 45% of the first-pass

[2] `joshua-decoder.org/fisher-callhome-corpus`

[3] LDC2010S01 and LDC2010T04

[4] LDC96S35 and LDC96T17

[5] Some online translation engines now provide optical-character recognition from images, reducing the potential effectiveness of this control for future work.

Source Data	Docs.	Segments	Spanish words	Translations	English words	Cost
Fisher (set one)	60	11,581	121,484	4	(avg) 118,176	$2,684
Fisher (set two)	759	138,819	1,503,003	1	1,440,727	$10,034
Callhome	120	20,875	204,112	1	201,760	$1,514
Combined	939	171,275	1,828,599	1	1,760,663	$14,232
Voting						+$1,433
Total						$15,665

Table 1: Corpus size and cost. Counts of segments and words were computed after pre-processing (§2).

Split	Words	Sentences
Fisher/Train	1,810,385	138,819
Dev	50,700	3,979
Dev2	47,946	3,961
Test	47,896	3,641
Callhome/Train	181,311	15,080
Devtest	47,045	3,966
Evltest	23,626	1,829
Europarl + NC	44,649,409	1,936,975

Table 2: Data splits for Fisher Spanish (top), Callhome Spanish (middle), and Europarl + News Commentary (bottom; for comparison). Words is the number of Spanish word tokens (after tokenization). The mean number of words per sentences ranges from 11.8 to 13.1.

translations. As a sanity check, we computed different accuracy thresholds for the voters, and the downstream ratings of the translators turned out to be relatively stable, so we were reasonably confident about the group of selected translators.

2.4. Complete Translations

The preferred translators were invited to translate the remaining Fisher data and all of the Callhome data at a higher wage, using the same strategy as the first round of translations. We obtained only one translation per utterance. Table 1 gives the size and cost of the entire translation corpus. To the best of our knowledge, the resulting corpus is the largest parallel dataset of audio, transcriptions, and translations. We anticipate that this data will be useful for research in a variety of cross-lingual speech applications, a number of which we explore ourselves in the following sections.

3. Collecting Speech Output

After collecting translations, we split the data into training, development, and test sets suitable for experimentation (Table 2). Callhome defines its own data splits, organized into train, devtest, and evltest, so we retained them. For Fisher, we produced four data splits: a large training section and three test sets (dev, dev2, and test). These test sets correspond to portions of the data where we have four translations.

The above procedures produced a three-way parallel corpus: Spanish audio, Spanish transcripts, and English translations. To this, we added speech recognizer output produced with the open-source Kaldi Automatic Speech Recognition System [18].[6]

In order to get output for the entire data set, we built multiple independent recognition systems:

- For Fisher/Dev2 and Fisher/Test, and all of the Callhome data, we used a recognition system built from Fisher/Train and tuned on Fisher/Dev.

- For Fisher/Train and Fisher/Dev, we used a 10-fold training and decoding scheme, where each fold was trained, tuned, and tested on a distinct 80/10/10 split. We then assembled these portions of the data set by taking the corresponding data from the test portions of these splits.

Each ASR system was built in the following manner. The phonetic lexicon included words from the training corpus, pronunciations for which were created using the LDC Spanish rule-based phonetic lexicon (LDC96L16). We then began with one round of monophone training, which was used for alignment and subsequent training with triphone Gaussian mixture models, which incorporated linear discriminant analysis with Maximum Likelihood Linear Transforms (MLLT) [19]. The results of triphone training were then used for Speaker Adaptive training [20, SAT]. Alignment and decoding for the SAT training step incorporated fMLLR [21]. We used a trigram language model derived solely from the training corpus and created with Kaldi tools.[7]

Along with the 1-best output, we extracted lattices representing the recognition hypotheses for each utterance. We applied epsilon-removal and weight-pushing to the lattices, and pruned them with a beam width of 2.0. All of these operations were performed using the OpenFST toolkit [22].

Finally, we also extracted and provide the oracle path from these lattices. These are useful in helping to quantify the missed performance in both the ASR and MT systems. Statistics about the lattices are presented in Table 3.

[6] `kaldi.sourceforge.net`

[7] The procedures, parameters, and design decisions of this process are captured in a custom Kaldi recipe, now distributed with Kaldi.

	WER		
	1-best	Oracle	# Paths
Fisher/Dev	41.3	19.3	28k
Fisher/Dev2	40.0	19.4	168k
Fisher/Test	36.5	16.1	48k
Callhome/Devtest	64.7	36.4	6,119k
Callhome/Evltest	65.3	37.9	1,328k

Table 3: Lattice statistics for the three Fisher and two Callhome test sets. Word error rates correspond to the 1-best and oracle paths from the lattice, and # Paths denotes the average number of distinct paths through each lattice. The average node density (the number of outgoing arcs) is 1.3 for Fisher and 1.4 for Callhome.

4. Experimental Setup

Our main interest is in the downstream performance of the MT system, and we report experiments varying different components of the ASR–MT pipeline to examine their effect on this goal. For Fisher, we use Dev for tuning the parameters of the MT system and present results on Dev2 (reserving Test for future use); for Callhome, we tune on Devtest and present results on Evltest. Because of our focus on speech translation, for all models, we strip all punctuation (except for contractions) from both sides of the parallel data.

For machine translation, we used Joshua, an open-source hierarchical machine translation toolkit written in Java [23]. Our grammars are hierarchical synchronous grammars [24]. Decoding proceeds by parsing the input with the source-side projection of the synchronous grammar using the CKY+ algorithm and combining target-side hypotheses with cube-pruning [24]. This algorithm can easily be extended to lattice decoding in a way that permits hierarchical decomposition and reordering of words on the input lattice [25].

The decoder's linear model comprises these features:

- Phrasal probabilities ($p(e|f)$ and $p(f|e)$)
- Lexical probabilities ($w(e|f)$ and $w(f|e)$)
- Rarity penalty, $\exp(1 - \text{count}(\text{rule}))$
- Word penalty
- Glue rule penalty
- Out-of-vocabulary word penalty
- 5-gram language model score
- Lattice weight (the input path's posterior log probability; where appropriate)

The language model is always constructed over the target side of the training data. These features are tuned using k-best batch MIRA [26], and results are reported on the average of three runs. Our metric is case-insensitive BLEU-4 [27] with four references (for Fisher) and one reference (for Callhome).

	Training set			
Interface	Euro	LDC	ASR	LDC +ASR
Transcript	41.8	58.7	54.6	58.7
1-best	24.3	35.4	34.7	35.5
Lattice	-	37.1	35.9	36.8
Oracle Path	32.1	46.2	44.3	46.3

Table 4: BLEU scores (four references) on Fisher/Dev2. The columns vary the data used to train the MT system, and the rows alter the interface between the ASR and MT systems.

	Training set			
Interface	Euro	LDC	ASR	LDC +ASR
Transcript	17.3	27.8	24.9	28.0
1-best	7.3	11.7	10.7	11.6
Lattice	-	12.3	11.5	12.3
Oracle Path	9.8	16.4	15.2	16.4

Table 5: BLEU scores (one reference) on Callhome/Evltest.

5. Experiments

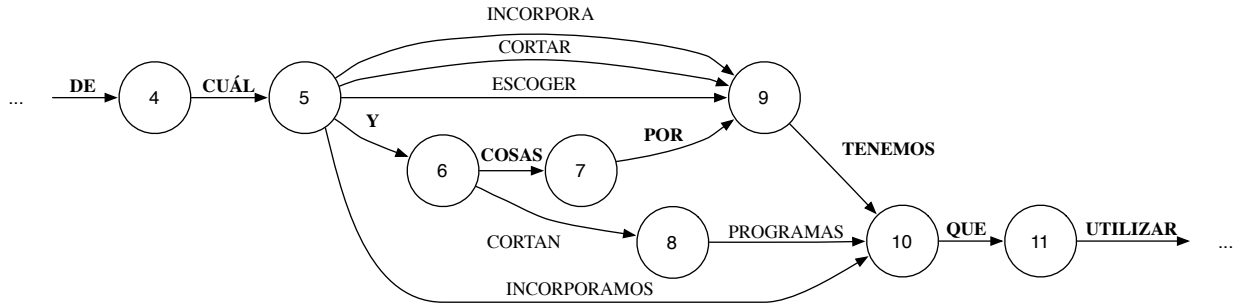
Our experiments largely center on an exploration varying one of two major components in the ASR–MT pipeline: (a) the training data used to build the machine translation engine, and (b) the interface between the ASR and MT systems.

For (a), we examine four training data sets (Table 2):

- *Euro*. The version 7 release of the Spanish-English Europarl dataset [17], a corpus of European parliamentary proceedings.
- *LDC*. An in-domain model constructed from paired LDC Spanish transcripts and their corresponding English translations, on Fisher Train, as described above.
- *ASR*. An in-domain model trained on pairs of Spanish ASR outputs and English translations.
- *LDC+ASR*. A model trained by concatenating the training data for LDC and ASR.

For (b), we vary the interface in four ways:

- *Transcript*. We translate the LDC transcripts. This serves as an upper bound on the possible performance.
- *1-best*. We translate the 1-best output as presented by the speech recognizer.
- *Lattices*. We pass a pruned lattice from the recognizer to the MT system.
- *Oracle Path*. The oracle path from the lattice, representing the best transcription found in the ASR system's hypothesis space (subject to pruning).



Transcript	*sí hablar de cuáles y cosas pero tenemos que utilizar la palabra matrimonio supongo*
1-best	sí habla de cuál incorporamos que utilizar la palabra matrimonio supongo
Lattice	sí habla de cuál escoger tenemos que utilizar la palabra matrimonio supongo
Reference	*yes [we can] talk about anything but we have to use the word marriage i guess*
1-best → MT	yes speaking of which incorporamos$_{\text{OOV}}$ to use the word marriage i suppose
Lattice → MT	yes speaking of which to choose we have to use the word marriage i suppose
1-best → Google	does speak of what we incorporate to use the word marriage guess

Figure 1: A subgraph of a lattice (sentence 17 of Fisher/Dev2) representing an ASR ambiguity. The oracle path is in bold. With access to the lattice, the MT system avoids the untranslatable word *incorporamos*, found in the 1-best output, producing a better translation. Above the line are inputs and the reference, with the *Lattice* line denoting the path selected by the MT system. The Google line is suggestive of the general difficulty in translating conversational speech.

Tables 4 and 5 contain results for the Fisher and Callhome datasets, respectively. The rest of this section is devoted to their analysis.

5.1. Varying the interface

The *Transcript* and *Oracle Path* interfaces represent upper bounds of different sorts. *Transcript* is roughly how well we could translate if we had perfect recognition, while *Oracle Path* is how well we could translate if the MT system could perfectly capitalize on the speech recognition lattice. From these baseline scores, it's clear that the quality of the speech recognition is the biggest hindrance to downstream machine translation quality, and therefore improving recognition accuracy qualifies as the best way to improve it.

However, there is significant room for MT improvement from the lattices themselves. Translating ASR lattices produces consistently better results than translating ASR 1-best output, corroborating an already well-attested finding for speech translation. Interestingly, these results hold true across the translation models, whether in-domain or out-of-domain, and when built from both LDC and ASR training data. It seems that the lattices truly contain paths that are better-suited to the translation engine, regardless of what was used to train the model. Figure 1 contains examples where lattice translation improves over translation of the ASR 1-best for this corpus.

In general, these numbers establish a relationship between word error rate and BLEU score. Figure 2 visualizes this relationship, by breaking out the data from Fisher/Dev and Fisher/Dev2 into its original twenty conversations, and plotting WER and BLEU for each of them.

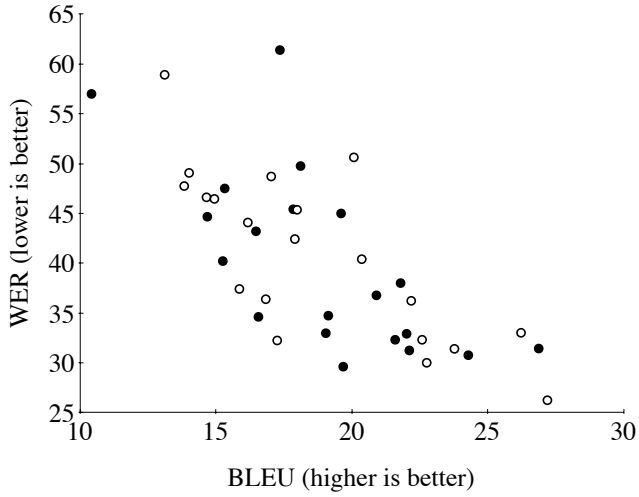


Figure 2: Conversation-level WER and BLEU, for conversations found in Fisher/Dev (open points) and Fisher/Dev2 (solid points). The Pearson's correlation coefficient is -0.72.

5.2. Varying the training data

The BLEU scores between columns 1 and 2 clearly demonstrate lessons well-known in the domain-adaptation literature. In our case, small, in-domain models built on the Fisher/Train significantly outperform the much larger (by a factor of twenty) but less relevant Europarl data. The test sentences in the Fisher and Callhome corpora, with their informal register and first-person speech, are a poor match for models trained on Parliamentary proceedings and news text.

While unsurprising, these results demonstrate the utility of the Fisher and Callhome Translation corpus for translating conversational speech, and are a further footnote on the conventional wisdom that "more data" is the best kind of data.

As an additional experiment, we tried building MT translation models from the Spanish ASR output (pairing the English translations with the ASR outputs instead of the Spanish LDC transcripts on Fisher/Train), based on the idea that errors made by the recognizer (between training and test data) might be regular enough that they could be captured by the translation system. Columns 3 and 4, which show worse BLEU scores than with the LDC translation model, provide preliminary evidence that this is not the case. This is not to claim that there is no utility to be found in training translation models on ASR output, but finding improvements from such will require something more than simply concatenating the two corpora.

6. Summary

We described the development and release of The Fisher and Callhome Spanish-English Speech Translation Corpus. The translations and ASR output (in the form of lattices and 1-best and oracle paths) complement their corresponding LDC acoustic data and transcripts, together producing a valuable dataset for research into the translation of informal Spanish conversational speech. This dataset is available from the Joshua website.[8]

7. References

[1] R. Zens, O. Bender, S. Hasan, S. Khadivi, E. Matusov, J. Xu, Y. Zhang, and H. Ney, "The RWTH phrase-based statistical machine translation system," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2005, pp. 155–162.

[2] E. Matusov, S. Kanthak, and H. Ney, "Integrating speech recognition and machine translation: Where do we stand?" in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 5. IEEE, 2006, pp. V–V.

[3] S. Novotney and C. Callison-Burch, "Cheap, fast and good enough: Automatic speech recognition with non-expert transcription," in *Proceedings of NAACL*, 2010.

[4] R. Zbib, E. Malchiodi, J. Devlin, D. Stallard, S. Matsoukas, R. Schwartz, J. Makhoul, O. Zaidan, and C. Callison-Burch, "Machine translation of Arabic dialects," in *Proceedings of NAACL-HLT*, 2012.

[5] M. Post, C. Callison-Burch, and M. Osborne, "Constructing parallel corpora for six indian languages via crowdsourcing," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 401–409.

[6] M. Eskenazi, G. Levow, H. Meng, G. Parent, and D. Suendermann, *Crowdsourcing for Speech Processing, Applications to Data Collection, Transcription and Assessment*. Wiley, 2013.

[7] V. Quan, M. Federico, and M. Cettolo, "Integrated n-best re-ranking for spoken language translation," in *Proceedings of Interspeech, Lisbon, Portugal*, 2005.

[8] N. Bertoldi, R. Zens, and M. Federico, "Speech translation by confusion network decoding," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4. IEEE, 2007, pp. IV–1297.

[9] S. Saleem, S. Jou, S. Vogel, and T. Schultz, "Using word lattice information for a tighter coupling in speech translation systems," in *Proc. Int. Conf. on Spoken Language Processing*, 2004, pp. 41–44.

[10] E. Matusov, S. Kanthak, and H. Ney, "On the integration of speech recognition and statistical machine translation," in *Proceedings of Interspeech, Lisbon, Portugal*, 2005.

[11] A. Sorokin and D. Forsyth, "Utility data annotation with Amazon Mechanical Turk," in *Proceedings of CVPR Workshops*, 2008.

[12] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of CHI*, 2008.

[13] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks," in *Proceedings of EMNLP*, 2008.

[14] C. Callison-Burch, "Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk," in *Proceedings of EMNLP*, 2009.

[15] G. Paolacci, J. Chandler, and P. G. Ipeirotis, "Running experiments on Amazon Mechanical Turk," *Judgment and Decision Making*, vol. 5, no. 5, pp. 411–419, 2010.

[16] O. F. Zaidan and C. Callison-Burch, "Crowdsourcing translation: Professional quality from non-professionals," in *Proceedings of ACL*, 2011.

[17] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Machine translation summit*, vol. 5, 2005.

[18] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and

[8] joshua-decoder.org/fisher-callhome-corpus

K. Vesely, “The Kaldi speech recognition toolkit,” in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, Dec. 2011, IEEE Catalog No.: CFP11SRW-USB.

[19] M. J. F. Gales, “Maximum likelihood linear transformations for HMM-Based speech recognition,” *Computer Speech and Language*, vol. 12, p. 75–98, 1998.

[20] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, “A compact model for speaker-adaptive training,” in *, Fourth International Conference on Spoken Language, 1996. ICSLP 96. Proceedings*, vol. 2, 1996, pp. 1137–1140 vol.2.

[21] C. Leggetter and P. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, Apr. 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230885700101

[22] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, “OpenFst: A general and efficient weighted finite-state transducer library,” *Implementation and Application of Automata*, pp. 11–23, 2007.

[23] M. Post, J. Ganitkevitch, L. Orland, J. Weese, Y. Cao, and C. Callison-Burch, “Joshua 5.0: Sparser, better, faster, server,” in *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August 2013, pp. 206–212.

[24] D. Chiang, “Hierarchical phrase-based translation,” *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[25] C. Dyer, S. Muresan, and P. Resnik, “Generalizing word lattice translation,” in *Proceedings of ACL*, Columbus, Ohio, June 2008, pp. 1012–1020.

[26] C. Cherry and G. Foster, “Batch tuning strategies for statistical machine translation,” in *Proceedings of NAACL-HLT*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 427–436.

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of ACL*, Philadelphia, Pennsylvania, USA, July 2002.

Unsupervised Learning of Bilingual Categories in Inversion Transduction Grammar Induction

Markus Saers Dekai Wu

Human Language Technology Center
Dept. of Computer Science and Engineering
Hong Kong University of Science and Technology
{masaers|dekai}@cs.ust.hk

Abstract

We present the first known experiments incorporating unsupervised bilingual nonterminal category learning within end-to-end fully unsupervised transduction grammar induction using matched training and testing models. Despite steady recent progress, such induction experiments until now have not allowed for learning differentiated nonterminal categories. We divide the learning into two stages: (1) a *bootstrap* stage that generates a large set of categorized short transduction rule hypotheses, and (2) a *minimum conditional description length* stage that simultaneously prunes away less useful short rule hypotheses, while also iteratively segmenting full sentence pairs into useful longer categorized transduction rules. We show that the second stage works better when the rule hypotheses have categories than when they do not, and that the proposed conditional description length approach combines the rules hypothesized by the two stages better than a mixture model does. We also show that the compact model learned during the second stage can be further improved by combining the result of different iterations in a mixture model. In total, we see a jump in BLEU score, from 17.53 for a standalone minimum description length baseline with no category learning, to 20.93 when incorporating category induction on a Chinese–English translation task.

1. Introduction

Even simple lexical translations are surprisingly context-dependent, in this paper we aim to learn a translation model that can base contextual translation decision on more than lexical n-grams, both in the input and output language. In a syntactic translation system such as inversion transduction grammars (ITGs), this can be achieved with unsupervised bilingual category induction. Surface-based and hierarchical models only use output language n-grams, and syntactic model typically choose the categories from either the input or the output language, or attempts to heuristically synthesize a set of bilingual categories from the two monolingual sets. In contrast, we attempt to learn a set of bilingual categories without supervision, which gives a unique opportunity to strike a good balance between the two approaches.

The specific translation of words and segments depend heavily on the context. A grammar-based translation model can model the context with nonterminal categories, which allows (a) moving beyond n-grams (as a compliment to the language model prior which is typically preserved), and (b) taking both the input and output language context into account. Typical syntactic MT systems either ignore categories (bracketing ITGs and hierarchical models), or derive the categories from tree-banks, which relies on choosing the set of categories from either language, or heuristically synthesize it from both; both approaches eliminates the full benefits of (b). In contrast, unsupervised induction of bilingual categories has the potential to take full advantage of (b).

Recent work has seen steady improvement in translation quality for completely unsupervised transduction grammar induction under end-to-end purely matched training and testing model conditions. In this paper, we take a further step along this line of research by incorporating unsupervised bilingual category induction into the learning process. To our knowledge, no previous attempt has been made to incorporate bilingual categories under such conditions. Matching the training and testing models as closely as possible is a fundamental principle taken for granted in most applications of ma-

[*] A full version of this paper appears at IWPT 2013.

chine learning, but for machine translation it has been the norm to see very different assumptions during training and testing, which makes it difficult to assess the effects of changing or tweaking the model—the observed effect may not be repeatable. By matching training and testing conditions, this risk is minimized.

A bilingual category is similar to a monolingual category in that it is realized as the left-hand side label of a (transduction) grammar rule, but differ in what it represents. A monolingual category only encodes how something relates to other parts of the language, a bilingual category should encode how a translation equivalence relates to other translation equivalences. It needs to account for the relationship between two languages as well as the relationship between the parts of the individual languages. This makes the usage of existing tagging schemes problematic. It would be possible to use the categories from either of the languages (assuming they are languages with enough resources) and impose these on the other language. This could work for closely related languages, but we are translating between English and Chinese: two very different languages, and we know that the category sets of either language is a poor fit for the other. Another possibility is to take the cross-product of the monolingual category sets, but handling such a large set of categories becomes unwieldy in ITG induction, a process which is resource intensive as is, without exploding the set of nonterminals. Instead, we opt for unsupervised learning of the bilingual categories during induction of the ITG itself.

The novel learning method we propose consists of an initial hypothesis generator that proposes (a) short lexical translations and (b) nonterminal categories, screened by a mechanism that (c) verifies the usefulness of the hypotheses while (d) uses them to further generate longer transduction rules. For convenience, our implementation breaks this into two stages: one that generates a large set of short transduction rule hypotheses, and another that iteratively segments long transduction rules (initialized from the sentence pairs in the training data) by trying to reuse a minimal subset of the hypotheses while chipping away at the long sentence pair rules until the conditional description length is minimized.

2. Background

Description length has been used before to drive iterative segmenting ITG learning [1]. We will use their algorithm as our baseline, but the simple mixture model we used then works poorly with our ITG with categories. Instead, we propose a tighter incorporation, where the rule segmenting learning is biased towards rules that are present in the categorized ITG.

We refer to this objective as minimizing *conditional description length*, since technically, the length of the ITG being segmented is *conditioned* on the categorized ITG. Conditional description length (CDL) is detailed in Section 3. The minimum CDL (MCDL) objective differs from the simple mixture model in that it separates the rule hypotheses into two groups: the ones that are used during segmentation and therefor carries over to the final induced ITG, and those that do not and are effectively filtered out. As we will see, MCDL far outperforms the mixture model when one of the ITGs has categories and the other does not.

A problem with the description length family of learning objectives is that they tend to commit to a segmentation when it would be wise to keep the unsegmented rule *as well*—a significant part of the success of phrase-based translation models comes from their approach to keep all possible segmental translations (that do not violate the prerequisite word alignment). We will show that we can counter this by combining different iterations of the same segmentation process into a single grammar, which gives a significant bump in BLEU.

By insisting on the fundamental machine learning principle of matching the training model to the testing model, we do forfeit the short term boost in BLEU that is typically seen when embedding a learned ITG in the midst of the common heuristics employed in statistical machine translation. For example, [2–14] all plug some aspect of the ITGs they learn into training pipelines for existing, mismatched decoders, typically in the form of the word alignment that an ITG imposes on a parallel corpus as it is biparsed. Our own past work has also taken similar approaches, but it is not necessary to do so—instead, any ITG can be used for decoding by directly parsing with the input sentence as a hard constraint, as we do in this paper. Although it allows you to tap into the vast engineering efforts that have gone into perfecting existing decoders, it also prevents you from surpassing them in the long run. The motivation for our present series of experiments is that, as a field we are well served by tackling the fundamental questions as well, and not exclusively focusing on engineering short term incremental BLEU score boosts where the quality of an induced ITG itself is obscured because it is embedded within many other heuristic algorithms.

When the structure of an ITG is induced without supervision, it is possible to get an effect that resembles MDL. [3] impose a sparsity prior over the rule probabilities to prevent the search from having to consider all the rules found in the Viterbi biparses. [4, 5, 8, 13, 14] use Gibbs sampling to learn ITGs with priors over the rule structures that serve a similar purpose to the model length component of description length. All of the above evaluate their models by biparsing the training data and feeding the imposed word alignment into an existing, mismatched SMT learning pipeline.

Transduction grammars can also be induced with supervision from treebanks, which cuts down the search space by enforcing external constraints [15]. Although this constitutes a way to borrow nonterminal categories that help the translation model, it complicates the learning process by adding external constraints that are bound to match the translation model poorly.

3. Conditional description length

Conditional description length (CDL) is a general method for evaluating a model and a dataset given a preexisting model. This makes it ideal for augmenting an existing model with a variant model of the same family. In this paper we will apply this to augment an existing inversion transduction grammar (ITG) with rules that are found with a different search strategy. CDL is similar to description length [16, 17], but the length calculations are subject to additional constraints. When minimum CDL (MCDL) is used as a learning objective, all the desired properties of minimum description length (MDL) are retained: the model is allowed to become less certain about the data provided that the it shrinks sufficiently to compensate for the loss in precision. MDL is a good way to prevent over-fitting, and MCDL retains this property, but for the task of inducing a model that is specifically tailored toward augmenting an existing model. Formally, the conditional description length is:

$$DL\left(\Phi, D|\Psi\right) = DL\left(D|\Phi, \Psi\right) + DL\left(\Phi|\Psi\right)$$

where Ψ is the fixed preexisting model, Φ is the model being induced, and D is the data. The total unconditional length is:

$$DL\left(\Psi, \Phi, D\right) = DL\left(D|\Phi, \Psi\right) + DL\left(\Phi|\Psi\right) + DL\left(\Psi\right)$$

In minimizing CDL, we fix $DL\left(\Psi\right)$ instead of allowing Ψ to vary as we would in full MCDL; to be precise, we seek:

$$\begin{aligned}
&\operatorname*{argmin}_{\Phi} DL\left(\Psi, \Phi, D\right) \\
&= \operatorname*{argmin}_{\Phi} DL\left(D|\Phi, \Psi\right) + DL\left(\Phi|\Psi\right) + DL\left(\Psi\right) \\
&= \operatorname*{argmin}_{\Phi} DL\left(\Phi, D|\Psi\right) \\
&= \operatorname*{argmin}_{\Phi} DL\left(D|\Phi, \Psi\right) + DL\left(\Phi|\Psi\right)
\end{aligned}$$

To measure the CDL of the data, we turn to information theory to count the number of bits needed to encode the data given the two models under an optimal encoding [18], which gives:

$$DL\left(D|\Phi, \Psi\right) = -\lg P\left(D|\Phi, \Psi\right)$$

To measure the CDL of the model, we borrow the encoding scheme for description length presented in [1], and define the conditional description length as:

$$DL\left(\Phi|\Psi\right) \equiv DL\left(\Phi\right) - DL\left(\Phi \cap \Psi\right)$$

To determine whether a model Φ has a shorter conditional description length, than another model Φ', it is sufficient to be able to subtract one length from the other. For the model length, this is trivial as we merely have to calculate the length of the difference between the two models in our theoretical encoding. For data length, we need to solve:

$$\begin{aligned}
&DL\left(D|\Phi', \Psi\right) - DL\left(D|\Phi, \Psi\right) \\
&\quad = -\lg P\left(D|\Phi', \Psi\right) - -\lg P\left(D|\Phi, \Psi\right) \\
&\quad = -\lg \frac{P\left(D|\Phi', \Psi\right)}{P\left(D|\Phi, \Psi\right)}
\end{aligned}$$

4. Generating rule hypotheses

In the first stage of our learning approach, we generate a large set of possible rules, from which the second stage will choose a small subset to keep. The goal of this stage is to keep the *recall* high with respect to a theoretical "optimal ITG", *precision* is achieved in the second stage. We rely on chunking and category splitting to generate this large set of rule hypotheses.

To generate these high-recall ITGs, we will follow the bootstrapping approach presented in [19], and start with a finite-state transduction grammar (FSTG), do the

chunking and category splitting within the FSTG framework before transferring the resulting grammar to a corresponding ITG. This is likely to produce an ITG that performs poorly on its own, but may be informative in the second stage.

5. Segmenting rules

In the second stage of our learning approach, we segment rules explicitly representing the entire training data, into smaller—more general—rules, reusing rules from the first stage whenever we can. By driving the segmentation-based learning with a minimum description length objective, we are learning a very concise ITG, and by conditioning the description length on the rules hypothesized in the first stage, we separate the good rule hypotheses from the bad: the good rules—along with their categorizing left-hand sides—are reused and the bad are not.

In this work, we are only considering segmentation of lexical rules, which keeps the ITG in normal form, greatly simplifying processing without altering the expressivity. A lexical ITG rule has the form $A \rightarrow e_{0..T}/f_{0..V}$, where A is the left-hand side nonterminal—the category, $e_{0..T}$ is a sequence of T (from position 0 up to but not including position T) L_0 tokens and $f_{0..V}$ is a sequence of V (from position 0 up to but not including position V) L_1 tokens. When segmenting this rule, three new rules are produced which take one of the following forms depending on whether the segmentation is inverted or not:

$$\begin{array}{rcl} A \rightarrow [BC] & & A \rightarrow \langle BC \rangle \\ B \rightarrow e_{0..S}/f_{0..U} & \text{or} & B \rightarrow e_{0..S}/f_{U..V} \\ C \rightarrow e_{S..T}/f_{U..V} & & C \rightarrow e_{S..T}/f_{0..U} \end{array}$$

All possible splits of the terminal rule can be accounted for by choosing the identities of B, C, S and U, as well as whether the split it straight or inverted.

The pseudocode for the iterative rule segmenting learning algorithm driven by minimal conditional description length can be found in Algorithm 1. It uses the methods `collect_biaffixes`, `eval_cdl`, `sort_by_delta` and `make_segmentations`. These methods collect all biaffixes in the rules of an ITG, evaluate the difference in conditional description length, sorts candidates by these differences, and commits to a given set of candidates, respectively. To evaluate the CDL of a proposed set of candidate segmentations, we need to calculate the difference in CDL between the current model, and the model that would result from committing to the candidate segmentations:

Algorithm 1 Iterative rule segmenting learning driven by minimum conditional description length.

```
Φ                                  ▷ The ITG being induced
Ψ                ▷ The ITG the learning is conditioned on
repeat
  δ_sum ← 0
  bs ← collect_biaffixes(Φ)
  bδ ← []
  for all b ∈ bs do
    δ ← eval_cdl(b, Ψ, Φ)
    if δ < 0 then
      bδ ← [bδ, ⟨b, δ⟩]
  sort_by_delta(bδ)
  for all ⟨b, δ⟩ ∈ bδ do
    δ' ← eval_cdl(b, Ψ, Φ)
    if δ' < 0 then
      Φ ← make_segmentations(b, Φ)
      δ_sum ← δ_sum + δ'
until δ_sum ≥ 0
return Φ
```

$$\begin{aligned} & DL\left(D, \Phi'|\Psi\right) - DL\left(D, \Phi|\Psi\right) \\ & \quad = DL\left(D|\Phi', \Psi\right) - DL\left(D|\Phi, \Psi\right) \\ & \qquad + DL\left(\Phi'|\Psi\right) - DL\left(\Phi|\Psi\right) \end{aligned}$$

The model lengths are trivial, as we merely have to encode the rules that are removed and inserted according to our encoding scheme and plug in the summed lengths in the above equation. This leaves the length of the data, which would be:

$$DL\left(D|\Phi', \Psi\right) - DL\left(D|\Phi, \Psi\right) = -\lg \frac{P\left(D|\Phi', \Psi\right)}{P\left(D|\Phi, \Psi\right)}$$

For the sake of convenience in efficiently calculating this probability, we make the assumption that:

$$P\left(D|\Phi, \Psi\right) \approx P\left(D|\Phi\right) = P\left(D|\theta\right)$$

where θ represents the model parameters, which reduces the difference in data CDL to:

$$-\lg \frac{P\left(D|\theta'\right)}{P\left(D|\theta\right)}$$

which lets us determine the probability through biparsing with the model being induced. Biparsing is, however, a very expensive operation, and we are making relatively small changes to the ITG, so we will further assume that we can estimate the CDL difference in closed

form based on the model parameters. Given that we are splitting the rule r_0 into the three rules r_1, r_2 and r_3, and that the probability mass of r_0 is distributed uniformly over the new rules, the new grammar parameters θ' will be identical to θ, except that:

$$\begin{aligned}
\theta'_{r_0} &= 0 \\
\theta'_{r_1} &= \theta_{r_1} + \frac{1}{3}\theta_{r_0} \\
\theta'_{r_2} &= \theta_{r_2} + \frac{1}{3}\theta_{r_0} \\
\theta'_{r_3} &= \theta_{r_3} + \frac{1}{3}\theta_{r_0}
\end{aligned}$$

We estimate the CDL of the corpus given this new parameters to be:

$$-\lg \frac{P\left(D|\theta'\right)}{P\left(D|\theta\right)} \approx -\lg \frac{\theta'_{r_1}\theta'_{r_2}\theta'_{r_3}}{\theta_{r_0}}$$

To generalize this to a set of rule segmentations, we construct the new parameters θ' to reflect all the changes in the set in a first pass, and then sum the differences in CDL for all the rule segmentations with the new parameters in a second pass.

6. Experimental setup

The learning approach we chose has two stages, and in this section we describe the different ways of using these two stages to arrive at a final ITG, and how we intend to evaluate the quality of those ITGs.

For the first stage, we will use the technique described in [19] to start with a finite-state transduction grammar (FSTG) and perform chunking before splitting the nonterminal categories and moving the FSTG into ITG form. We perform one round of chunking, and two rounds of category splitting (resulting in 4 nonterminals and 4 preterminals, which becomes 8 nonterminals in the ITG form). At each stage, we run a few iterations of expectation maximization using the algorithm detailed in [20] for biparsing. For comparison we also bootstrap a comparable ITG that has not had the categories split. Before using either of the bootstrapped ITGs, we eliminate all rules that do not have a probability above a threshold that we fixed to 10^{-50}. This eliminates the highly unlikely rules from the ITG.

For the second stage, we use the iterative rule segmentation learning algorithm driven by minimum conditional description length that we introduced in Section 5. We will try three different variants on this algorithm: one without an ITG to condition on, one conditioned on the chunked ITG, and one conditioned on the chunked ITG with categories. The first variant is completely independent from the chunked ITGs, so we will also try to create mixture models with it and the chunked ITGs.

Since the MCDL objective tends to segment large rules and count on them being recreatable when needed, many of the longer rules that would be good to have when translating are not explicitly in the grammar. This is potentially a source of translation mistakes, and to investigate this, we create a mixture model from iterations of the segmenting learning process leading up to the learned ITG.

All the above outlined ITGs are trained using the IWSLT07 Chinese–English data set [21], which contains 46,867 sentence pairs of training data, and 489 Chinese sentences with 6 English reference translations each as test data; all the sentences are taken from the traveling domain. Since the Chinese is written without whitespace, we use a tool that tries to clump characters together into more "word like" sequences [22].

To test the learned ITGs, we use them as translation systems with our in-house ITG decoder. The decoder uses a CKY-style parsing algorithm [23–25] and cube pruning [26] to integrate the language model scores. For language model, we use a trigram language model trained with the SRILM toolkit [27] on the English side of the training corpus. To evaluate the resulting translations, we use BLEU [28] and NIST [29].

7. Results

In this section we present the empirical results: bilingual categories help translation quality under the experimental conditions detailed in the previous section. The results are summarized in Table 1. As predicted the base *chunked only* ITG fares poorly, while the categories help a great deal in the *chunked w/categories only* ITG—though the scores are not very reliable when in this low range.

The trade-off between model and data size during segmentation conditioned on the ITG with categories is illustrated in Figure 1. It starts out with most of the total description being used to describe the model, and very little to describe the data. This is the degenerate situation where every sentence pair is its own lexical rule. Then there is a sharp drop in model size with a slight increase in data size. This is where the most dramatic generalizations take place. It levels off fairly quickly, and the minor adjustments that take place on the plateau still

Table 1: Experimental results. *Chunked* is the base model, which has categories added to produce *chunked w/categories*. *Segmented* corresponds to the second learning stage, which can be done in isolation (*only*), *mixed with* a base model, or *conditioned on* a base model.

Model	**BLEU**	**NIST**	**Categories**
Chunked ITG only	3.76	0.0119	1
Chunked ITG w/categories only	9.39	0.7481	8
Segmented ITG only	17.53	4.5409	1
Segmented ITG mixed with chunked ITG	10.23	0.2886	1
Segmented ITG mixed with chunked ITG w/categories	12.06	1.1415	8
Segmented ITG conditioned on chunked ITG	17.04	4.4920	1
Segmented ITG conditioned on chunked ITG w/categories	19.02	4.6079	8
... with iterations combined	20.20	4.8287	8
... and improved search parameters	20.93	4.8426	8

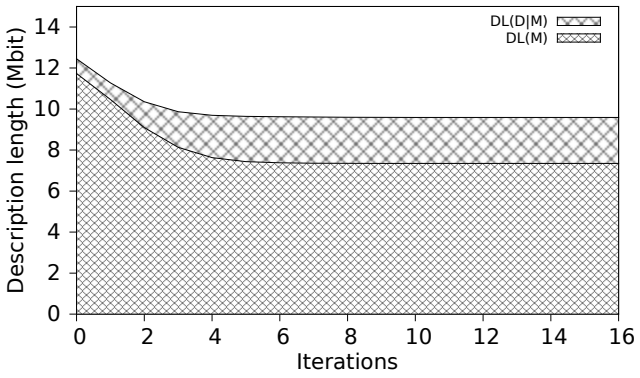


Figure 1: Description length in bits over the different iterations of segmenting search. The lower part represents model CDL, $DL\left(\Phi|\Psi\right)$, and the upper part represents data CDL, $DL\left(D|\Phi,\Psi\right)$.

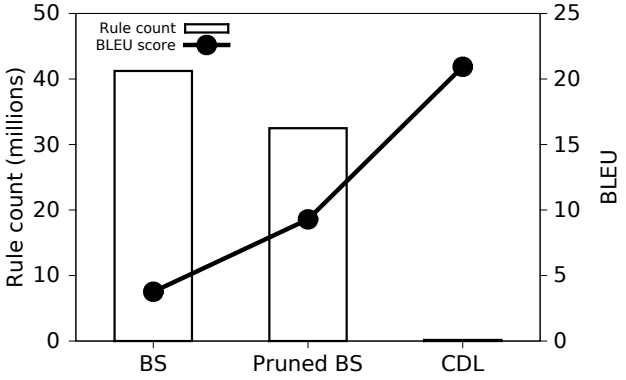


Figure 2: Rule count versus BLEU scores for the bootstrapped ITG, the pruned bootstrapped ITG and the segmented ITG conditioned on the pruned bootstrapped ITG.

represent valid generalizations, they just have a very small effect on the over-all description length of either the model or the data.

That the chunked ITG with split categories suffers from having too many irrelevant rules is clearly seen in Figure 2, where we plotted the number of rules contrasted to the BLEU score. Merely pruning to a threshold helps somewhat, but the sharper improvement—both in terms of model size and BLEU score—is seen with the filtering that MCDL represents.

A number of interesting lessons emerge from the results, as follows.

7.1. Minimum CDL outperforms mixture modeling

The segmenting approach works as expected (*segmented only*), essentially reproducing the results reported by [1] for this style of bilingual grammar induction. Interestingly, however, where they had success with the mixture model combining the base ITGs with the ITG learned through the segmenting approach (*segmented mixed with...*), we see a significant drop in translation quality. This may be because we have categories in our base ITG and they do not.

7.2. Category induction strongly improves minimum CDL learning

When we use the base ITGs to condition the segmenting approach, we see something interesting. The base ITG that has categories causes a sharp 1.5 BLEU point rise in translation quality (compare *segmented only* to *segmented conditioned on chunked w/categories*).

In contrast, the base ITG that does not have cate-

gories causes a slight 0.5 BLEU point *fall* in translation quality (compare *segmented only* to *segmented conditioned on chunked*).

7.3. Redundant segmental rule granularities help

As mentioned, the minimum description length objective may be theoretically nice, but it also relies on the learned ITG being able to reassemble segmented rules with fairly high fidelity at decoding time. To demand that all transduction rules are reduced to exactly a single right level of granularity may be a bit of a tall order.

Our way to test this was to uniformly mix the ITGs at different iterations though the segmenting process. By mixing the ITG after each iteration up to the one labeled *segmented conditioned on chunked w/categories*, we get the same model labeled *...with iterations combined*, which secures an additional 1.18 BLEU points.

7.4. Tuning search parameters

Lastly, for the best approach, we further experimented with adjusting the parameters somewhat. Pruning the base grammar harder (a threshold of 10^{-10} instead of 10^{-50}), and allowing for a wider beam (100 items instead of 25) during the parsing part of the segmenting learning approach, we see the BLEU score rise to 20.93.

7.5. Analysis of learned rules

A manual inspection of the content of the categories learned reveals that the main nonterminal contains mainly structural rules, segments that it could not segment further. The latter type of rules varies from full clauses such as that ' s a really beautiful dress/真是件漂亮的衣服 to reasonable translation units such as Kazuo Yamada/Ｋａｚｕｏ Ｙａｍａｄａ, which is really hard to capture because each Latin character on the Chinese side is its own individual token whereas the English side has whole names as individual tokens.

A second nonterminal category contains punctuation such as full stop and question mark, along with , sir/，先生, which can be considered as a form of punctuation in the domain of the training data.

A third nonterminal category contains personal pronouns in subject form (I, we, he, and also ambiguous pronouns that could be either subject or object form such as you and it) paired up with their respective Chinese translations. It also contains please/请, which—like pronouns in subject form—occurs frequently in the beginning of sentence pairs.

A fourth nonterminal category contains pairs such as can/吗, do you/吗, is/吗, could you/吗 and will you/吗 — instances where Chinese typically makes a statement, possibly eliding the pronoun, and adds the question particle (吗) to the end, and where English prefixes that statement with a verb; both languages use a question mark in the particular training data we used. The main nonterminal learned that this category typically was used in inverted rules, and the other translation equivalences conform to that pattern. They include where/在哪, where the Chinese more literally translates to on/at which, what/什么 which is a good translation, and have/了, where the English auxiliary verb corresponds well to the Chinese particle signaling *perfect aspect*.

Other categories appear to be consolidating, with a mix of nouns, verbs, adjectives, and adverbials. Chinese words and phrases typically can function as any of these, so it is possible that differentiating them may require increased emphasis on the English half of the rules.

Although the well-formed categories are few and somewhat trivial, it is very encouraging to see them emerging without any form of human supervision. Future work will expand to continue learning an even wider range of categories.

8. Conclusions

We have presented the first known experiments for incorporating bilingual category learning within completely unsupervised transduction grammar induction under end-to-end matched training and testing model conditions. The novel approach employs iterative rule segmenting driven by a minimum conditional description length learning objective, conditioned on a prior defined by a stochastic ITG containing automatically induced bilingual categories. We showed that this learning objective is superior to the previously used mixture model, when bilingual categories are involved. We also showed that the segmenting learning algorithm may be committing too greedily to segmentations since combining the ITGs with different degrees of segmentation gives better scores than any single point in the segmentation process; this points out an interesting avenue of future research. We further saw that the segmenting minimization of conditional description length can pick up some of the signal in categorization that was buried in noise in the base ITG the induction was conditioned on, leading to an ITG with much clearer categories. In total

we have seen an improvement of 3.40 BLEU points due to the incorporation of unsupervised category induction.

9. Acknowledgements

This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under BOLT contract no. HR0011-12-C-0016, and GALE contract nos. HR0011-06-C-0022 and HR0011-06-C-0023; by the European Union under the FP7 grant agreement no. 287658; and by the Hong Kong Research Grants Council (RGC) research grants GRF620811, GRF621008, and GRF612806. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the EU, or RGC.

10. References

[1] Markus Saers, Karteek Addanki, and Dekai Wu, "Combining top-down and bottom-up search for unsupervised induction of transduction grammars," in *SSST-7*, Atlanta, GA, Jun 2013, pp. 48–57.

[2] Colin Cherry and Dekang Lin, "Inversion transduction grammar for joint phrasal translation modeling," in *SSST*, Rochester, NY, Apr 2007, pp. 17–24.

[3] Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea, "Bayesian learning of non-compositional phrases with synchronous parsing," in *ACL-08: HLT*, Columbus, OH, Jun 2008, pp. 97–105.

[4] Phil Blunsom, Trevor Cohn, and Miles Osborne, "Bayesian synchronous grammar induction," in *NIPS 21*, Vancouver, Canada, Dec 2008.

[5] Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne, "A Gibbs sampler for phrasal synchronous grammar induction," in *ACL-IJCNLP 2009*, Suntec, Singapore, Aug 2009, pp. 782–790.

[6] Aria Haghighi, John Blitzer, John DeNero, and Dan Klein, "Better word alignments with supervised ITG models," in *ACL-IJCNLP 2009*, Suntec, Singapore, Aug 2009, pp. 923–931.

[7] Markus Saers and Dekai Wu, "Improving phrase-based translation via word alignments from stochastic inversion transduction grammars," in *SSST-3*, Boulder, CO, Jun 2009, pp. 28–36.

[8] Phil Blunsom and Trevor Cohn, "Inducing synchronous grammars with slice sampling," in *NAACL HLT 2010*, Los Angeles, CA, Jun 2010, pp. 238–241.

[9] David Burkett, John Blitzer, and Dan Klein, "Joint parsing and alignment with weakly synchronized grammars," in *NAACL HLT 2010*, Los Angeles, CA, Jun 2010, pp. 127–135.

[10] Jason Riesa and Daniel Marcu, "Hierarchical search for word alignment," in *ACL 2010*, Uppsala, Sweden, Jul 2010, pp. 157–166.

[11] Markus Saers, Joakim Nivre, and Dekai Wu, "Word alignment with stochastic bracketing linear inversion transduction grammar," in *NAACL HLT 2010*, Los Angeles, CA, Jun 2010, pp. 341–344.

[12] Markus Saers and Dekai Wu, "Principled induction of phrasal bilexica," in *EAMT-2011*, Leuven, Belgium, May 2011, pp. 313–320.

[13] Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara, "An unsupervised model for joint phrase alignment and extraction," in *ACL HLT 2011*, Portland, OR, Jun 2011, pp. 632–641.

[14] Graham Neubig, Taro Watanabe, Shinsuke Mori, and Tatsuya Kawahara, "Machine translation without words through substring alignment," in *ACL 2012*, Jeju Island, Korea, Jul 2012, pp. 165–174.

[15] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer, "Scalable inference and training of context-rich syntactic translation models," in *COLING/ACL 2006*, Sydney, Australia, Jul 2006, pp. 961–968.

[16] Ray J. Solomonoff, "A new method for discovering the grammars of phrase structure languages," in *IFIP*, 1959, pp. 285–289.

[17] Jorma Rissanen, "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics*, vol. 11, no. 2, pp. 416–431, Jun 1983.

[18] Claude Elwood Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, Jul, Oct 1948.

[19] Markus Saers, Karteek Addanki, and Dekai Wu, "From finite-state to inversion transductions: Toward unsupervised bilingual grammar induction," in *COLING 2012*, Mumbai, India, Dec 2012, pp. 2325–2340.

[20] Markus Saers, Joakim Nivre, and Dekai Wu, "Learning stochastic bracketing inversion transduction grammars with a cubic time biparsing algorithm," in *IWPT'09*, Paris, France, Oct 2009, pp. 29–32.

[21] C. S. Fordyce, "Overview of the IWSLT 2007 evaluation campaign," in *IWSLT 2007*, 2007, pp. 1–12.

[22] Zhibiao Wu, "LDC Chinese segmenter," 1999. [Online]. Available: http://www,lde.upenn,edu/Projects/Chinese/segmenter/mansegment,

[23] John Cocke, *Programming languages and their compilers: Preliminary notes*. Courant Institute of Mathematical Sciences, New York University, 1969.

[24] Tadao Kasami, "An efficient recognition and syntax analysis algorithm for context-free languages," Air Force Cambridge Research Laboratory, Tech. Rep. AFCRL-65-00143, 1965.

[25] Daniel H. Younger, "Recognition and parsing of context-free languages in time n^3," *Information and Control*, vol. 10, no. 2, pp. 189–208, 1967.

[26] David Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[27] Andreas Stolcke, "SRILM – an extensible language modeling toolkit," in *ICSLP2002 - INTERSPEECH 2002*, Denver, CO, Sep 2002, pp. 901–904.

[28] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "BLEU: a method for automatic evaluation of machine translation," in *ACL-02*, Philadelphia, PA, Jul 2002, pp. 311–318.

[29] George Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *HLT '02*, San Diego, CA, 2002, pp. 138–145.

A Study in Greedy Oracle Improvement of Translation Hypotheses

Benjamin Marie[1,2]*, Aurélien Max*[1,3]

(1) LIMSI-CNRS, Orsay, France
(2) Lingua et Machina, Le Chesnay, France
(3) Univ. Paris Sud, Orsay, France
{firstname.lastname}@limsi.fr

Abstract

This paper describes a study of translation hypotheses that can be obtained by iterative, greedy oracle improvement from the best hypothesis of a state-of-the-art phrase-based Statistical Machine Translation system. The factors that we consider include the influence of the rewriting operations, target languages, and training data sizes. Analysis of our results provide new insights into some previously unanswered questions, which include the reachability of previously unreachable hypotheses *via* indirect translation (thanks to the introduction of a `rewrite` operation on the source text), and the potential translation performance of systems relying on pruned phrase tables.

1. Introduction

There are two opposing ways in which one may look at the current level of performance reached by Statistical Machine Translation (SMT) systems. One is that the results of SMT systems are still quite unreliable and not appropriate for dissemination or even post-editing by human translators, in particular for low-resourced and/or difficult language pairs, and for situations where domain adaptation is difficult. The other, opposing view is that some contexts allow SMT systems to reach very high performance, including when large enough quantities of adapted data are available, e.g. by using SMT systems in conjunction with translation memories, which has yielded much interest into the study and use of human post-editing and tools for supporting this activity.

Such performance levels typically correspond to the utilization of the *best* translation hypothesis produced by a given system, which is a reflection of the system's relative evaluation of the translations in its search space. Previous oracle studies have shown that the best attainable performance of such systems was in fact much higher than their best output [1]. This is achieved by relaxing pruning and reordering constraints imposed on decoders, and maximizing some evaluation metrics score rather than the system's own scoring function. Such studies are useful, in particular, to make explicit the potential of a given system configuration (training data, extraction procedures, etc.) and to possibly exhibit the difficult parts of a source text (e.g. [2])) as well as the possible defects of reference translations. A lesson that can be drawn from these results is the poor adequacy of the internal scores of translation quality used by current systems.

Another interesting potential use of oracle studies is that they can produce useful data under the form of individual post-editing steps that may be used to improve existing translation hypotheses. Initial attempts at *automatic post-editing* of SMT output approached the problem as one of second-pass translation between automatic predictions and correct translations [3]. Among the drawbacks of such approaches, large quantities of texts have to be translated to learn post-editing models, which are then furthermore specific to a given version of a given system and consequently not straightforwardly reusable. Some large collections of manually revised translations have been collected [4, 5], which can be used e.g. for sub-sentential confidence estimation. However, such data sets are costly to acquire, in particular for some language pairs, and may again be, on some aspects, too specific to a given version of the MT system used.

In this article, we describe an approach to build a related resource, but for a modest cost and with possibly wider applicability. We resort to greedy rewriting of translation hypotheses, in a similar spirit to Langlais et al. [6], to find the sequence of rewriting steps which maximizes the quality of translation hypotheses with respect to some evaluation metrics and reference translation(s). Individual rewritings are based on the repertoire of biphrases units of some phrase-based SMT systems, and thus do not have to correspond to plausible rewritings made by human translators.

While we aim to use such a resource to learn to identify improvable fragments (e.g. [4]) and learn discriminative rerankers (e.g. [7]), we will here focus on a systematic study of such an artificial resource. Our experiments will study the following factors:

- rewriting operations: we will use a revised and extended set of previously used operations [6], and introduce an original operation which allows source sentence rewriting (`rewrite`), as well as a target phrase deletion operation (`remove`);
- training data size: we will use 5 different sizes of training data, where training data are split independently from their relation to the test data;

- number of available reference translations: we will be able to verify whether phenomena observed when a single reference translation is available can also be observed when as many as 7 reference translations allow for a much more robust evaluation of translation quality;

- phrase table filtering: we will use unfiltered phrase tables and phrase tables filtered using a significance testing criterion [8];

- target language: we will use French as the source language, and 10 other European languages as target languages, with exactly the same training data;

- beam size: finally, we will also consider various beam sizes to get some account of the quantity of search errors made by our greedy decoder, although this aspect is not central to the present study.

The remainder of this article is organized as follows. Section 2 introduces greedy oracle decoding and describes the operations that we have used in this work. Section 3 presents our choice of data, systems, and search settings for this work. Our experiments are then detailed in Section 4. We finally summarize our main findings and present some of our future work in Section 5.

2. Greedy oracle decoding

Greedy decoding for Statistical Machine Translation was introduced in [9], as a fast solution to the NP-complete problem of finding the best translation hypothesis from a translation engine's search space.[1] Although such a technique was shown to produce more search errors than its dynamic programming-based counterpart for max-derivation approximation, Langlais et al. [6] described an implementation of greedy search decoding that could improve the best hypothesis from a then state-of-the-art DP-decoder. Subsequent work using a Gibbs sampler for approximating maximum translation decoding [10] showed, however, the adequacy of the approximations made by recent decoders for finding the best translation in their search space, leaving as the main source to account for current translation performance the scoring of translation hypotheses.

Our objective in the present work is not to improve the decoder score of the translation hypotheses that are found, but rather to obtain, by construction, iteratively better hypotheses by using a sentence-level measure of actual translation performance (hence, some approximation of an *oracle*). The sub-optimality of the search is not a problem for our purpose, so we resort to a straightforward greedy algorithm to build such sequences of iteratively improving translation hypotheses.

[1] An optimal, but more costly solution, relying on integer programming, was also proposed in the same article.

Algorithm 1 Greedy oracle search algorithm

Require: $source$ (input sentence), $beamSize$

$nbest \leftarrow$ NBEST_LIST$(source, beamSize)$
$oneBest \leftarrow$ GET_ONE_BEST$(nbest)$
loop
  $newNbestList \leftarrow$ INITIALIZE_LIST()
  $sCurrent \leftarrow$ sBLEU$(oneBest)$
  $s \leftarrow sCurrent$
  for all $h \in$ NEIGHBORHOOD_BEAM$(nbest)$ **do**
    $c \leftarrow$ sBLEU(h)
    $newNbestList \leftarrow$ ADD$(h, c, beamSize)$
    if $c > s$ **then**
      $s \leftarrow c$
    end if
  end for
  if $s = sCurrent$ **then**
    return $oneBest$
  else
    $nbest \leftarrow newNbestList$
    $oneBest \leftarrow$ GET_ONE_BEST$(newNbestList)$
  end if
end loop

Our greedy oracle decoding is illustrated as pseudo-code in Algorithm 1. We take as seeds the n-best, segmented translation hypotheses of a phrase-based SMT system. At each iteration, a number of best hypotheses relative to our evaluation metrics are kept in a beam until convergence is obtained. Each surviving hypothesis undergoes a number of modifications by means of a repertoire of rewriting operations on bi-phrases that define a neighborhood function. We used the following operations (N denotes the number of biphrases, T the maximum number of entries per source phrase in a translation table, R the maximum number of entries per source phrase in a source rewriting table, and S the average number of tokens per source phrase)[2]:

1. `replace` ($\mathcal{O}(N.T)$): replaces the translation of a source phrase with another translation from the phrase table;

2. `split` ($\mathcal{O}(N.S.T^2)$): splits a source phrase into all possible sets of two (contiguous) phrases, and uses `replace` on each of the resulting phrases;

3. `merge` ($\mathcal{O}(T.N)$): merges two contiguous source phrases and uses `replace` on the resulting new phrase;

4. `move` ($\mathcal{O}(N^2)$): moves the target phrase of a biphrase to all inter-phrase positions in the translation hypothesis;

[2] Complexity is expressed in terms of the maximum number of hypotheses that will be considered given a seed hypothesis. Note that some of our operations have a much higher complexity than those in [6], which is justified by the fact that we want to explore a larger search space.

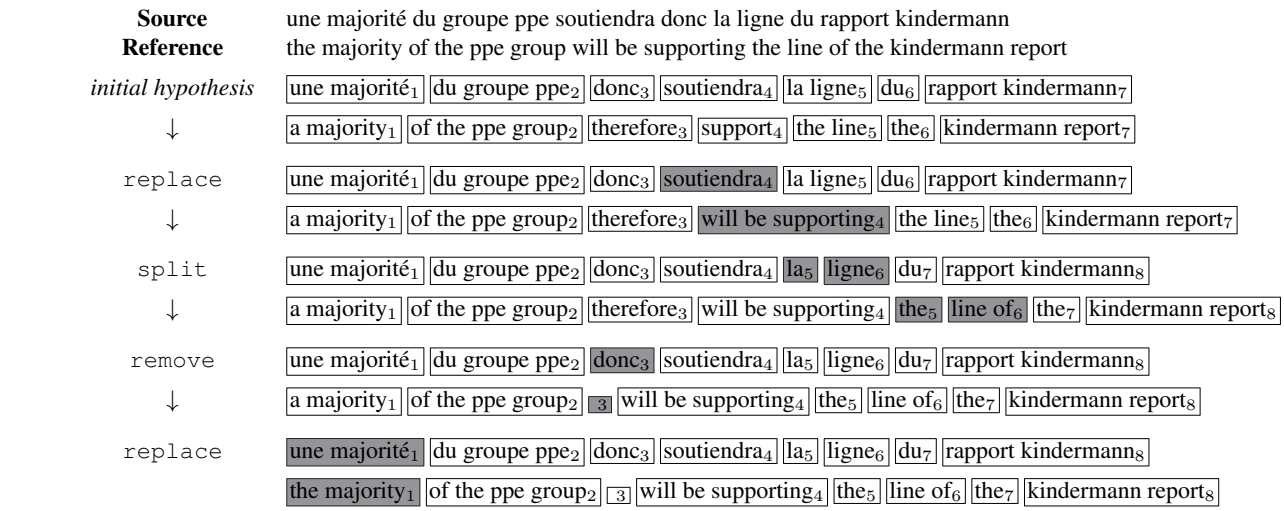


Figure 1: Trace of an example greedy oracle decoding between French and English. The final state is reached after a sequence of 4 operations (`replace`, `split`, `remove`, `replace`). Indices in the frames around phrases indicate bilingual alignments originating from the seed hypothesis produced by the `Moses` decoder.

5. `remove` ($\mathcal{O}(N)$): deletes the translation of a given biphrase (which remains available as a placeholder for later rewritings);

6. `rewrite` ($\mathcal{O}(N.R)$): replaces the source phrase of a biphrase with some other source phrase, and replaces its translation with the translations of this new source phrase; note that, by construction, we only need to put in the source rewriting table biphrases that allow to reach n-grams that are not reachable using other operations.

Such a greedy oracle decoder has several limitations. As said previously, it cannot perform a full exploration of the search space and will consequently make search errors; we will report in Section 4 some effects of beam size. Furthermore, our operations are applied on some bilingual phrase segmentation of the source sentence and the translation hypothesis, and `split` and `merge` operations will only allow to visit a subset of all rewritings that would be licenced if considering word alignments only. However, this is acceptable for our purpose, as a subsequent objective will be to improve the output of a state-of-the-art phrase-based system using a repertoire of such phrase-based rewriting operations.

One may also keep in mind that some increases in translation scores will not always correspond to actual improvements as judged by human translators. Indeed, some attempts at maximizing a single metrics will result in inappropriate transformations, such as arbitrarily removing words or moving them to positions where e.g. they do not break any longer substrings from the reference translation. One solution may be to make use of a mixture of complementary translation metrics, which may however make computation much more expensive; we leave this to our future work, accepting for now the fact that important metrics score differences (e.g. up to 37 BLEU points and 31 TER points for French to English translation in this study) should always correspond to a majority of clear improvements.

Figure 1 shows an example of a trace by our system of iterative improvement of a translation from French into English, starting from a competitive initial hypothesis (see section 3). A local maximum is here reached after 4 rewriting operations. Examples for the other types of rewriting operations are shown on Figure 2.

3. Experimental settings

In order to experiment with several target languages under the same conditions, we used the Europarl corpus of parliamentary debates[3], and computed the intersection for 11 languages using English as pivot. From the collected data, we extracted held-out, later entries as tuning and test sets (see Table 1). We used French as our sole source language, and experimented with all other possible target languages. English was used as the main target language of the study, notably in settings where the training data was reduced to smaller fractions. Furthermore, in order to verify how our oracles would behave in situations where the evaluation metrics could make use of several possible reference translations, we also used the BTEC corpus of basic traveling expressions [11], allowing us to use 16 references for tuning our baseline systems and 7 references for evaluating them on the French to English language pair (see Table 1).

We built state-of-the-art phrase-based SMT systems using the open source `Moses` system[4], using standard settings and models and MERT [12] for optimizing the parameters on the tuning set. Trigrams target language models were es-

[3] `http://statmt.org/europarl/`
[4] `http://www.statmt.org/moses`

previous	... le projet qui ferait gagner le plus de temps sur un ferroviaire$_{15}$ trajet$_{16}$ très long$_{17}$
	... the project which would win the more time on a rail$_{15}$ route$_{16}$ very long$_{17}$
`move`	... le projet qui ferait gagner le plus de temps sur un ferroviaire trajet très long
	... the project which would win the more time on a very long rail route
previous	il est évident que parler d' intermodalité présuppose un profond changement de la culture d' entreprise .
	it is clear that speak intermodality presupposes a profound change in the business culture .
`rewrite`	il est évident que débat d' intermodalité présuppose un profond changement de la culture d' entreprise .
	it is clear that discussion on intermodality presupposes a profound change in the business culture .
previous	qu' il me soit$_3$ permis$_4$ dès lors de le placer dans une perspective plus historique .
	it would therefore be$_3$ allowed$_4$ to put it into a more a more historical perspective .
`merge`	qu' il me soit permis dès lors de le placer dans une perspective plus historique .
	it would therefore be permitted to put it into a more a more historical perspective .

Figure 2: Examples of applications of rewriting operations not already illustrated on the trace of Figure 1.

	train	**tune**	**test**		
	# M-tok.	# K-tok.	# K-tok.	BLEU	TER
Europarl corpora					
fr	10.2	32.8	32.8	-	-
en	8.8	28.3	28.6	29.1	54.0
/2	4.4	|	|	28.6	54.4
/4	2.2	|	|	27.6	55.4
/8	1.1	|	|	26.1	56.8
/16	0.5	|	|	25.2	58.4
da	8.4	27.0	27.2	23.2	61.3
de	8.4	27.1	27.1	17.0	68.0
el	8.8	28.5	28.5	23.5	62.2
es	9.2	29.5	29.7	35.9	49.7
fi	6.4	20.6	20.5	11.2	79.7
it	10.2	28.9	29.0	31.6	55.3
nl	8.9	28.2	28.7	21.2	64.6
pt	9.1	29.4	29.3	33.4	52.8
sv	7.9	25.7	25.8	21.0	62.7
BTEC corpus					
fr	0.2	0.5	0.5	-	-
en	0.2	0.5*	0.5**	59.6	24.6

Table 1: Top: Statistics for our Europarl training (up to 310K bi-sentences), tune (1K bi-sentences) and test (1K bi-sentences) corpora. Translation performance is given for all baseline systems using French as the source language. Bottom: Statistics for our BTEC training, tuning (16 references*) and test (7 references**) corpora.

timated from the bilingual training data only, using Kneser-Ney smoothing. Results for all baseline systems and all training conditions are reported in Table 1, using BLEU and TER as complementary indicators of translation performance.

We used the greedy search operations described in Section 2. We implemented various approximations to speed up decoding. In particular, we limited candidate replacements for `replace`, `split` and `merge` to phrases that contain at least one token in common with the reference translation, except for the 50 most frequent tokens.[5] We used sentence-level smoothed BLEU [13] as our objective function for greedy decoding (using a single (Europarl) or several reference translations (BTEC)), but will use corpus-level BLEU and individual n-gram precisions, as well as TER, to report translation performance.

4. Experiments and analysis

4.1. Rewriting operations

Using our main language pair, French to English, we experimented with each individual rewriting operation, as well as with the full set; see Table 2. The two operations that individually lead to the largest improvements are not surprisingly those that have access to replacement translations from the phrase table, `replace` and `split`. The larger improvements with the latter are due to the combination of sub-`replace` operations, which encompass translations attainable by composition as well as possibly more combinations not seen associated with the larger source phrases. Conversely, `merge` is of moderate use, but still manages to capture some cases where translations cannot be obtained by composition. As the sole operation, `remove` has almost no impact on translation, and may in fact only artificially inflate low-order *n*-gram precision values. `move` has a moderate impact, not too surprisingly more apparent on BLEU and higher-order *n*-gram precision than on TER, which may be attributed in part to the language pair (see Section 4.3). The impact of the `rewrite` operation will be specifically discussed in section 4.4.

[5] Although lowering this value led to fewer search errors, we deemed the chosen value a good compromise time-wise.

	Europarl fr→en (1 ref.)							BTEC fr→en (7 refs.)						
	BLEU					TER	avg #	BLEU					TER	avg #
	score	1g	2g	3g	4g	score	**iterations**	score	1g	2g	3g	4g	score	**iterations**
baseline	29.0	63.2	35.5	22.6	14.6	54.0	-	59.62	85.08	67.13	53.33	41.48	24.60	-
beam size = 1														
`merge`	31.8	65.3	38.3	25.2	16.9	51.7	0.75	60.43	85.43	67.84	54.32	42.35	24.32	0.07
`move`	32.0	63.2	39.1	25.8	17.3	53.3	1.01	61.70	85.08	69.52	55.84	43.87	24.60	0.16
`remove`	29.7	67.1	39.2	25.6	16.9	50.0	1.03	59.62	85.08	67.13	53.33	41.48	24.60	0.00
`replace`	42.1	73.9	48.8	34.8	25.1	42.5	4.40	66.50	88.72	73.40	60.90	49.33	23.67	0.91
`rewrite`	29.8	64.5	36.2	23.0	14.0	53.5	0.38	59.69	85.07	67.12	53.48	41.57	24.68	0.04
`split`	45.7	74.3	52.7	39.1	28.6	41.3	4.46	69.34	88.05	75.36	64.62	53.90	27.07	1.24
all	66.5	88.2	73.8	62.6	53.0	23.1	11.04	77.30	91.17	81.67	73.78	64.98	23.47	1.92
beam size = 2														
all	66.6	88.1	73.9	62.8	53.2	23.0	11.19	77.88	91.44	82.36	74.37	65.68	23.16	2.28
beam size = 5														
all	67.8	88.5	74.9	64.3	55.0	22.3	11.26	79.06	91.88	83.29	75.67	67.47	22.94	2.12

Table 2: Effects of individual operations and beam size (left: Europarl; right: BTEC).

Potential improvements to translation hypotheses using the original phrase table are very large. However, this may not reflect accurately *actual improvements*. One important reason for this is the fact that a single reference translation usually does not represent all the acceptable wordings of a translation. Looking at the BTEC condition, where the baseline evaluated on 7 reference translations is much stronger than in the Europarl condition, we still find significant increases in BLEU score with a relative contribution of operations that is well correlated to that obtained on the more difficult, single-reference Europarl condition. The main source of improvement for translation hypotheses thus resides in translating using generally smaller phrases (`split`) and choosing more appropriate translations for phrases (`replace`).

Next, we look at when each operation is used when they are all activated. The distribution of operations on Europarl is given on Figure 3 by looking at operations from each quarter of complete sequences (thus each corresponding to an average of $11.04/4 = 2.76$ operations). The first quarter of operations, yielding almost half of the full improvement, mostly consists of alternative translations (`split` and `replace`). The `move` operation contributes more after the initial burst of operations, while `remove` progressively acts on phrases for which `split` cannot propose any further improvement from the reached hypotheses.

All subsequent experiments will be conducted with a beam size of 1 to limit computation time.[6] Table 2 additionally provides results for larger beams, which gives some account of the reduction in search errors corresponding to a larger number of iterations per sentence (on average, there is 0.22 more iteration per sentence using a beam of 5, but at the cost of a running time multiplied by a factor of more than 3).

[6]On a single core of a 2.2Ghz machine with 64Gb memory, decoding our whole test sets took roughly 6 hours for a beam size of 1, 8 hours for a beam size of 2, and more than 20 hours for a beam size of 5.

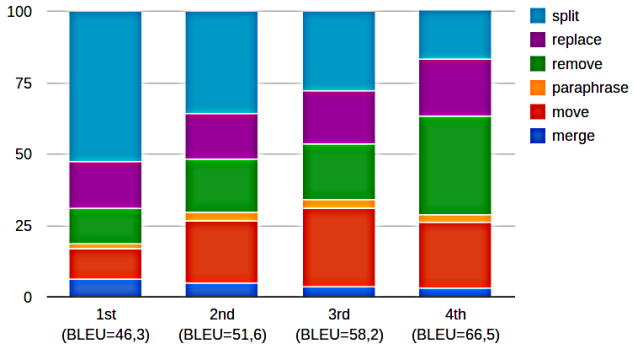


Figure 3: Distribution of types of operations per quarters of operations during greedy oracle search. Corresponding BLEU scores obtained after each quarter of iterations are indicated on the legend.

4.2. Training data size and phrase table filtering

Predicting translation performance given the available amount of training data is a useful problem [14]. Here, we look at how much training data size impacts the performance attainable by our oracle decoder. We reduce training data size up to 16 times on the Europarl condition, without selecting data in any way relative to the dev and test set. Results are given in Table 3. Whereas reducing by half the quantity of training data roughly corresponds to the loss of 1 BLEU point or less, we find that loss in oracle performance, although also regular for each training data size reduction, is close to 5 BLEU points. This fact may be often overlooked in the SMT research community, where it is commonly known that doubling the size of the training data typically has only a small impact on translation performance. Our results show that this is mostly a result of the limitations of the scoring function used by decoders, and that attainable improvements benefit much more from the added training data.

A related question is whether pruned phrase tables, which can yield competitive translation performance while retaining only small fractions of the original phrase table entries, would be significantly different in terms of attainable translations. We used the widely used significance pruning of Johnson et al. [8], and selected a configuration where phrase pairs occurring once in the bilingual corpus and composed from phrases also occurring once on their respective side of the corpus (so-called 1-1-1 configurations) are pruned. Looking at the results on Table 3, we find that keeping only 27% of the original phrase table entries indeed yielded no loss in translation performance at rank 1 for the decoder. Although the intuitions for filtering such phrase pairs include the fact that they may correspond to noise or offer too little reusability, the important drop in oracle performance (-11.2 BLEU points and +8.8 TER points) clearly indicates that a significant part of the filtered entries, although apparently poorly scored by the translation system, would have in fact largely benefited the system.[7]

4.3. Target languages

Classes of language pairs correspond to very different challenges for SMT systems, as exemplified by the large-scale study reported in [15]. In this set of experiments, we wanted to assess oracle performance for a number of target languages with various types of relationship to the source language (e.g. closely related (Spanish), completely unrelated (Finnish), different sentence structure (German), etc.) Results are shown in Table 4 for the 10 target languages of our study in the Europarl condition. Relative improvements in BLEU scores range from roughly +100% (for Spanish and Portuguese) to more than +300% (Finnish). This latter case seems particularly instructive: although not directly comparable to the absolute values reached for other target languages, phrase tables do contain entries that can significantly improve automatic translation into such a complex language as Finnish.[8] We observe, in particular, a very large increase in n-gram precision at all sizes.

Another interesting result concerns romance target languages, which obtain both the smallest relative increase in BLEU (around +100%) and the largest relative reduction in TER (up to -63%). Our hypothesis to account for this fact is that the improvements on n-gram precisions do not result in the strongest increases overall in BLEU, but that given that many such improvements for long target phrases are indeed possible, this globally results in sentence orderings that are more symmetric between oracle outputs and reference translations.

We further look at the distributions of rewriting operations per target language, given on Figure 4. `replace` operations appear uniformely useful for all languages, illustrating the relative inadequacy of the translation models used by the decoders across languages. `split` operations are more numerous for target languages with good baseline performance (e.g. English and Portuguese). This can be attributed to some over-confidence in long bi-phrases that can be extracted from the training data, which not always permit to attain the expected reference translation. Conversely, we note slightly more `merge` operations for romance languages and Greek, a fact that should be investigated further. While phrases used by the decoder used should be generally shorter, a significant number of source fragments are nonetheless inaccurately translated compositionally when their correct translation is available.[9]

Not surprisingly, we also note a larger use of `move` operations for translating into German (and, to a lesser extent, Dutch and Scandinavian languages). Likewise, we find, at no surprise, that Finnish required a more important number of deletions of target words associated to source phrases, a reflection of the much compositional morphology of the language, which makes capturing appropriate biphrases difficult when such a language is involved.

		BLEU		TER		#. iterations
		score	+rew	score	+rew	avg. per sent.
da	*baseline*	23.2		61.3		-
	oracle	58.4	+0.9	29.5	-0.8	10.7
de	*baseline*	17.0		68.0		-
	oracle	55.1	+1.4	32.0	-1.2	13.3
el	*baseline*	23.5		62.2		-
	oracle	62.8	+1.0	26.5	-0.6	11.5
en	*baseline*	29.0		54.0		-
	oracle	66.5	+0.6	23.1	-0.4	11.0
es	*baseline*	35.9		49.7		-
	oracle	74.0	+0.5	18.2	-0.5	10.7
fi	*baseline*	11.2		79.7		-
	oracle	46.1	+1.2	38.1	-1.2	11.3
it	*baseline*	31.6		55.2		-
	oracle	71.2	+1.1	20.4	-1.7	11.3
nl	*baseline*	21.2		64.6		-
	oracle	56.3	+1.6	32.4	-0.7	12.9
pt	*baseline*	33.4		52.8		-
	oracle	69.8	+0.7	21.5	-0.5	10.2
sv	*baseline*	21.0		62.7		-
	oracle	59.9	+1.0	27.8	-1.1	11.2

Table 4: Effects of target language (Europarl). '`+rew(rite)`' indicates the specific contribution of the corresponding improvement (in BLEU or TER) of the oracle score.

[7] We note, however, that using a filtered phrase table already yields an interesting level of oracle translation improvement, with a very modest running time (less than half an hour on a single core for decoding the 1000 sentences of our test set).

[8] We must, however, acknowledge the fact that the target language model used for baseline decoding could not be very competitive here, which is particularly true for this target language.

[9] Among other possibilities, a stronger language model may help correct this to some extent. In this study, priority was put on ensuring that all systems were built from the same data.

	BLEU						TER		#. biphrases
	baseline	oracle	1g	2g	3g	4g	baseline	oracle	in phrase table
`full`	29.1	65.9	87.9	73.3	61.9	52.4	54.0	23.5	735,273
/2	28.6	60.8	85.5	68.8	56.5	46.4	54.4	27.5	419,716
/4	27.6	55.6	82.8	64.3	51.0	40.7	55.4	31.1	239,647
/8	26.1	51.1	79.8	60.0	46.3	36.0	56.8	35.1	137,719
/16	25.2	46.0	76.9	55.2	41.1	30.7	58.4	39.0	79,837
`sigtest`	29.1	54.7	81.4	63.0	50.3	40.4	54.1	32.3	203,672

Table 3: Effects of training data size and phrase table filtering (all operations but `rewrite`) (Europarl).

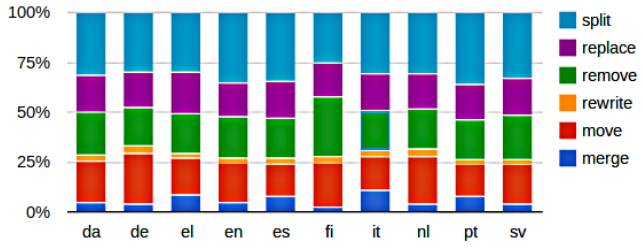


Figure 4: Distribution of operations per target language.

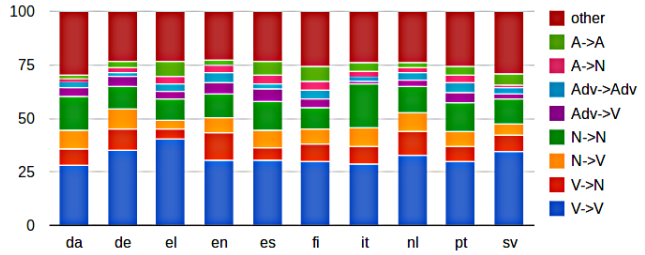


Figure 5: Distribution of main part-of-speech patterns of source `rewrite` for translation from French.

4.4. Reachability of new reference fragments

Our `rewrite` operation allows to reach fragments from the reference translations that are not directly reachable using `replace` only. Using this operation alone for French to English translation on the Europarl condition (Table 2) led to an improvement of +0.8 BLEU and -0.5 TER, for an average number of 0.38 applications per sentence decoding. Results across target languages (Table 4) show that languages that benefit the most from this increased reachability (more than +1 BLEU and -1 TER) mostly corresponds to languages with lower baseline scores, indicating that alignment difficulty (considering that the exact same training data were used for all language pairs) is responsible to some extent.

Positive applications of such an operation, as previously proposed by [16, 17] using source paraphrase lattices, include a large typology of configurations largely not limited to strict paraphrase phenomena, as illustrated on Figure 5. For instance, using English as the source language for illustration purposes, correctly translating the English word *buying* (in *not by buying other countries' quotas*) by *rachat* (in the expected translation *non par le rachat du "droit à polluer" d'un autre pays*) can only be done by translating the noun *purchase* instead. Studying source rewriting patterns on part-of-speeches (see Table 5) shows that French, with a rich verbal inflection system, mostly requires rewriting of verbs into verbs, with significantly fewer cases for nouns into nouns, and fewer yet for adjectives into adjectives. The most represented types with a change of category are verbs into nouns, nouns into verbs, and adverbs into verbs.

source	**reference**	**`rewrite` phrases**
abused	dénaturé	different
buying	rachat	purchase
complex	multitude	number | series | wealth
damaging	désastreuse	disastrous
drivers	des personnels	people
excuse	argument	argument | grounds | reason

Table 5: Examples of English source rewritings (note that English was used as source language here for illustration purposes) and their new reachable French reference translation fragment.

5. Conclusion

This article has presented a study of iteratively improved translation hypotheses, starting from competitive baseline hypotheses up to translation hypotheses of very high quality, even for comparatively difficult language pairs. Although we implemented a non-optimal solution to finding the hypotheses that maximize a single automatic metrics score, several useful facts were empirically demonstrated. Our study first confirmed the important potential for improvement of current phrase-based SMT systems, both in situations where a single or several reference translations are available, and the difficulty of the translation scoring problem. Such conclusions naturally pave the way for further research in discriminatively training systems, more particularly based on dynamic reranking using so-called pseudo-references [7], by focusing more particularly on the rewriting of possibly ill-translated phrases [2, 4].

We have also made explicit the relative contribution of a number of rewriting operations, including an original one, `rewrite`, which allows us to turn around the common acceptance that unique reference translations are poor representations of acceptable translations, and to claim that the specificities of a unique source text sometimes are responsible for (automatic) translation difficulty. Previously, Schroeder et al. [16] had shown the potential of using many human rewritings of input texts, and Khalilov and Sima'an [18] had shown the potential of using reorderings of input texts, but to our knowledge this work is the first to focus on the contribution of local indirect translation.[10] Paraphrasing the training data [19, 20] in a carefull manner is one way to provide access to such knowledge during translation.

Other salient results of our study include the empirical demonstration that pruned phrase tables significantly limit the potential of SMT systems, and that current SMT systems have the potential to already produce very good translation hypotheses even for difficult language pairs, however difficult this may be to achieve in practice. Part of our intended future work will focus on identifying high-quality greedy sequences of rewriting operations, and to compare them to edit sequences made by human post-editors, for whom finding a close-to-shortest route to translation improvement can be difficult.

6. Acknowledgements

This work was partially supported by ANR project Trace (ANR-09-CORD-023) and by a grant from LIMSI.

7. References

[1] G. Wisniewski, A. Allauzen, and F. Yvon, "Assessing Phrase-Based Translation Models with Oracle Decoding," in *EMNLP*, Cambridge, USA, 2010.

[2] B. Mohit and R. Hwa, "Localization of Difficult-to-Translate Phrases," in *WMT*, Prague, Czech Republic, 2007.

[3] M. Simard, C. Goutte, and P. Isabelle, "Statistical Phrase-based Post-editing," in *NAACL*, Rochester, USA, 2007.

[4] N. Bach, F. Huang, and Y. Al-Onaizan, "Goodness: A Method for Measuring Machine Translation Confidence," in *ACL*, Portland, USA, 2011.

[5] M. Potet, E. Esperança-Rodier, L. Besacier, and H. Blanchon, "Collection of a Large Database of French-English SMT Output Corrections," in *LREC*, Istanbul, Turkey, 2012.

[10] We should, of course, repeat such experiments using several reference translations and larger training data sets.

[6] P. Langlais, A. Patry, and F. Gotti, "A Greedy Decoder for Phrase-Based Statistical Machine Translation," in *TMI*, Skovde, Sweden, 2007.

[7] P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar, "An End-to-End Discriminative Approach to Machine Translation," in *ACL*, Sydney, Australia, 2006.

[8] H. Johnson, J. Martin, G. Foster, and R. Kuhn, "Improving Translation Quality by Discarding Most of the Phrasetable," in *EMNLP*, Prague, Czech Republic, 2007.

[9] U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada, "Fast decoding and optimal decoding for machine translation," in *ACL*, Toulouse, France, 2001.

[10] A. Arun, P. Blunsom, C. Dyer, A. Lopez, B. Haddow, and P. Koehn, "Monte Carlo inference and maximization for phrase-based translation," in *CoNLL*, Boulder, USA, 2010.

[11] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, "Toward a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversations in the Real World," in *LREC*, Las Palmas, Spain, 2002.

[12] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *ACL*, Sapporo, Japan, 2003.

[13] C.-Y. Lin and F. J. Och, "ORANGE: a Method for Evaluating Automatic Evaluation Metrics for Machine Translation," in *COLING*, Geneva, Switzerland, 2004.

[14] P. Kolachina, N. Cancedda, M. Dymetman, and S. Venkatapathy, "Prediction of Learning Curves in Machine Translation," in *ACL*, Jeju, Korea, 2012.

[15] P. Koehn, A. Birch, and R. Steinberger, "462 machine translation systems for Europe," in *MT Summit*, Ottawa, Canada, 2009.

[16] J. Schroeder, T. Cohn, and P. Koehn, "Word Lattices for Multi-Source Translation," in *EACL*, Athens, Greece, 2009.

[17] T. Onishi, M. Utiyama, and E. Sumita, "Paraphrase Lattice for Statistical Machine Translation," in *ACL, short papers*, Upsala, Sweden, 2010.

[18] M. Khalilov and K. Sima'an, "Statistical Translation After Source Reordering: Oracles, Context-Aware Models, and Empirical Analysis," *Natural Language Engineering*, vol. 18, no. 4, pp. 491–519, 2012.

[19] A. Max, "Example-based Paraphrasing for Improved Phrase-Based Statistical Machine Translation," in *EMNLP*, Cambridge, USA, 2010.

[20] W. He, S. Zhao, H. Wang, and T. Liu, "Enriching SMT Training Data via Paraphrasing," in *IJCNLP*, Chiang Mai, Thailand, 2011.

Source-Error Aware Phrase-Based Decoding for Robust Conversational Spoken Language Translation

Sankaranarayanan Ananthakrishnan, Wei Chen, Rohit Kumar, and Dennis Mehay

Speech, Language, and Multimedia Business Unit
Raytheon BBN Technologies
Cambridge, MA 02138, U.S.A.
{sanantha,wchen,rkumar,dmehay}@bbn.com

Abstract

Spoken language translation (SLT) systems typically follow a pipeline architecture, in which the best automatic speech recognition (ASR) hypothesis of an input utterance is fed into a statistical machine translation (SMT) system. Conversational speech often generates unrecoverable ASR errors owing to its rich vocabulary (e.g. out-of-vocabulary (OOV) named entities). In this paper, we study the possibility of alleviating the impact of unrecoverable ASR errors on translation performance by minimizing the contextual effects of incorrect source words in target hypotheses. Our approach is driven by locally-derived penalties applied to bilingual phrase pairs as well as target language model (LM) likelihoods in the vicinity of source errors. With oracle word error labels on an OOV word-rich English-to-Iraqi Arabic translation task, we show statistically significant relative improvements of 3.2% BLEU and 2.0% METEOR over an error-agnostic baseline SMT system. We then investigate the impact of imperfect source error labels on error-aware translation performance. Simulation experiments reveal that modest translation improvements are to be gained with this approach even when the source error labels are noisy.

1. Introduction

Conversational speech translation enables monolingual speakers of different languages to communicate with one another. The pipeline consists of ASR transcription of the input source language utterance, followed by text-to-text translation by SMT, and optional text-to-speech synthesis (TTS) in the target language. ASR performance is often a crucial bottleneck in the performance of speech translation systems, because it has a significant downstream impact on the SMT component. This is an important issue especially for spontaneous conversational speech, which exhibits a rich vocabulary even in domain-constrained applications, often resulting in a high OOV word rate. In the force protection and medical assistance domains, targeted under the DARPA TransTac and BOLT programs, a significant fraction of OOV entities refer to names of people, places, organizations, and objects. These OOV entities cause acoustically similar in-vocabulary words that best fit the linguistic context to be substituted in the 1-best ASR transcription, as illustrated in Figure 1. Furthermore, ASR errors caused by OOV entities are *unrecoverable*, i.e. there is no path in the ASR lattice that corresponds to the correct transcription.

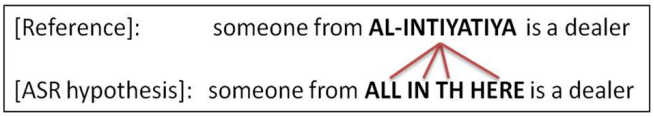


Figure 1: Unrecoverable ASR misrecognition caused by an OOV named-entity.

Translation errors caused directly by unrecoverable ASR errors, e.g. due to translation of source words substituted or inserted in place of an OOV entity, are unavoidable. However, these unrecoverable source language errors also affect translations of surrounding regions of error-free source words due to contextual effects. The goal of error-aware translation is to minimize the contextual impact of source errors and obtain the best possible translation for the correctly recognized portions of the utterance. We study this possibility by modifying a phrase-based SMT decoder to include penalties for bilingual phrase pairs spanning erroneous and error-free regions of input, and target language model (LM) likelihoods in the vicinity of source

errors. The proposed features are naturally integrated within a standard log-linear phrase-based translation model, resulting in a straightforward development and tuning process.

The remainder of this paper is organized as follows. Section 2 presents an overview of related work in this area. Section 3 describes the baseline speech translation pipeline, including details on the ASR and SMT systems. A detailed description of the proposed error-aware SMT decoding approach is given in Section 4. Experimental results are presented in Section 5. Finally, Section 6 concludes this paper with a brief discussion of our contribution and presents directions for future research in this area.

2. Relation to prior work

Integration of ASR and MT has gained popularity in the SLT community as a way of improving translation performance with potentially noisy input. This ranges from simple ASR post-processing to obtain segment boundaries or to insert punctuation [1, 2] to more sophisticated techniques such as joint decoding [3] and/or augmenting the SMT search space with ASR n-best lists, lattices, or word graphs (confusion networks) [4, 5]. The latter approach relies on the fact that the n-best list or lattice might contain a better hypothesis that could generate a more accurate translation. However, it is of limited utility in improving translation performance for utterances that generate unrecoverable ASR errors. Furthermore, the joint search space can be very large, making it difficult to implement some of these approaches for low memory, small form-factor devices that are preferred for SLT applications.

Our proposed approach is inspired by the idea of *attention-shift decoding* for ASR [6], where an input utterance is comprised of reliable *islands* and unreliable *gaps*. In this framework, initial hypotheses are constructed for the islands, and used to fill in the intermediate gaps in conjunction with additional information sources. In the case of SLT, islands refer to correctly recognized segments of the input utterance, while gaps consist of unrecoverable ASR errors. Our goal is to maximize translation performance on the correct islands, while minimizing interference from the incorrect gaps. In the SLT task domain, gaps will always generate translation errors and can only be filled in through additional external input (e.g. clarification dialog with the user). We refer the reader to our previous work [7] for more details on some of these interactive methods. In this paper, we focus solely on improving translation performance on the islands.

Distribution Statement "A" (Approved for Public Release, Distribution Unlimited)

3. Baseline systems

The baseline ASR and SMT systems for our SLT application are built on data from the DARPA TransTac English-Iraqi Arabic parallel two-way spoken dialogue collection. These data span a variety of domains including force protection (e.g. checkpoint, reconnaissance, patrol), medical diagnosis and aid, maintenance and infrastructure, etc., and are conversational in genre. We focused on the English-to-Iraqi Arabic direction because this was a primary requirement of the ongoing DARPA BOLT program, under which a significant part of this research was conducted.

The baseline English ASR was based on the BBN Byblos system, which uses a multi-pass decoding strategy where models of increasing complexity are used in successive passes in order to refine the recognition hypotheses [8]. The acoustic model was trained on approximately 200 hours of transcribed English speech from the TransTac corpus. The LM was trained on 5.8M English sentences (60M words), drawn from both in-domain and out-of-domain sources. LM and decoding parameters were tuned on a held-out development set of 3,534 utterances (45k words). With a dictionary of 38k words, we obtained 12.8% WER on a separate held-out test set of 3,138 utterances.

Our English-to-Iraqi Arabic SMT system was trained on a parallel corpus derived from the TransTac collection (773k sentence pairs, 7.3M words). Phrase pairs were extracted from bidirectional IBM Model 4 word alignment [9, 10] based on the heuristic approach of [11]. The target LM was trained on Iraqi Arabic transcriptions from the parallel corpus. Our phrase-based decoder (similar to Moses [12]) performs beam search stack decoding based on a standard log-linear model, whose parameters were tuned with MERT [13] on a held-out development set (3,534 sentence pairs, 45k words). The BLEU and METEOR scores of this system on a noise-free held-out test set (3,138 sentence pairs, 38k words) were 16.1 and 42.5, respectively.

4. Error-aware SMT decoding

Phrase-based SMT decoders rely on context in order to construct a reasonably fluent translation of an input source sentence. Local source context is captured by multi-word phrase pairs, while local target context is modeled both by phrase pairs as well as a n-gram target LM. By definition, error regions in source input (gaps) produce incorrect translations. This affects translation of surrounding regions of error-free input (islands) due to two primary contextual effects:

1. Selection of phrase pairs whose source phrases span islands and gaps, leading to mixing of correct and incorrect words in the source context.

2. Erroneous target LM history causing propagation of bad hypotheses at the boundaries between translations of source gaps and islands.

Our proposed approach to error-aware phrase-based SMT decoding involves minimizing the contextual impact gaps can have on the translation of islands. We encourage this separation between translation of islands and gaps in two different ways: (a) by discouraging the decoder from choosing phrase translation pairs whose source phrases span island-gap boundaries; and (b) by preventing the propagation of bad target hypotheses generated by source gaps through the application of dynamic target language model penalties.

Throughout this paper, we assume that each ASR-hypothesized source word s_i is tagged with a corresponding probability of error e_i, ranging from 0.0 (correct) to 1.0 (error). These error probabilities might be based on oracle error labels (e.g. Levenshtein alignment of ASR transcription with the reference), or automatically estimated through some machine learning inference process. In interactive spoken language translation systems, source error information may also be gleaned directly from the user through clarification techniques such as ASR confirmation [7]. In the latter approach, the user hears a synthesized version of the ASR 1-best hypothesis, and can inform the system of incorrect regions (gaps) in the hypothesis.

We introduce two new features that leverage source error probabilities to minimize gap interference in translation of islands. These features are evaluated at run-time and integrate directly within the log-linear translation model framework. Tunable parameter weights associated with these features can be optimized with MERT on an appropriate development set. The proposed approach is highly efficient because it preserves the original search space and adds virtually no complexity to the SMT decoder.

4.1. Phrase pair error span penalty

We introduce a penalty term that applies to phrase pairs whose source phrases span the boundary between an island and a gap, thereby discouraging selection of erroneous source contexts for translation of correctly recognized words. This also encourages separation of incorrect target words generated by gaps from correct hypotheses due to islands, permitting replacement with other information that can render the translation comprehensible. For instance, the interactive SLT system described in [7] automatically identifies source gaps generated by OOV named entities, and replaces incorrect target words due to them with an audio segment corresponding to the spoken name.

The error span penalty is evaluated at run-time for each candidate phrase pair in the search graph based on the source words it spans. It is computed as the maximal difference between error probabilities of successive constituent words in the source phrase, and applies equally to all translation options generated by that source phrase.

$$F_{X \rightarrow Y}(s_i, s_j) = - \max_{i \leq k < j} \mid e_{k+1} - e_k \mid \qquad (1)$$

Equation 1 illustrates the evaluation of this feature for a sample phrase pair $X \rightarrow Y$ which spans contiguous source words $(s_i, \ldots, s_j)$ with error probabilities $(e_i, \ldots, e_j)$. The rationale behind this feature is that source phrases spanning island-gap boundaries are likely to exhibit large internal differences in source error probability. The error span penalty discourages the decoder from choosing translations whose source phrases potentially span island-gap boundaries. However, it does not penalize phrase pairs that exclusively span either correct source words (islands) or incorrect source words (gaps).

4.2. Target language model penalty

Bad phrase translations generated by source gaps can negatively influence the target context through the n-gram target LM. To prevent the propagation of errors in this manner, we introduce a dynamic target LM penalty that is applied to each translation hypothesis in the

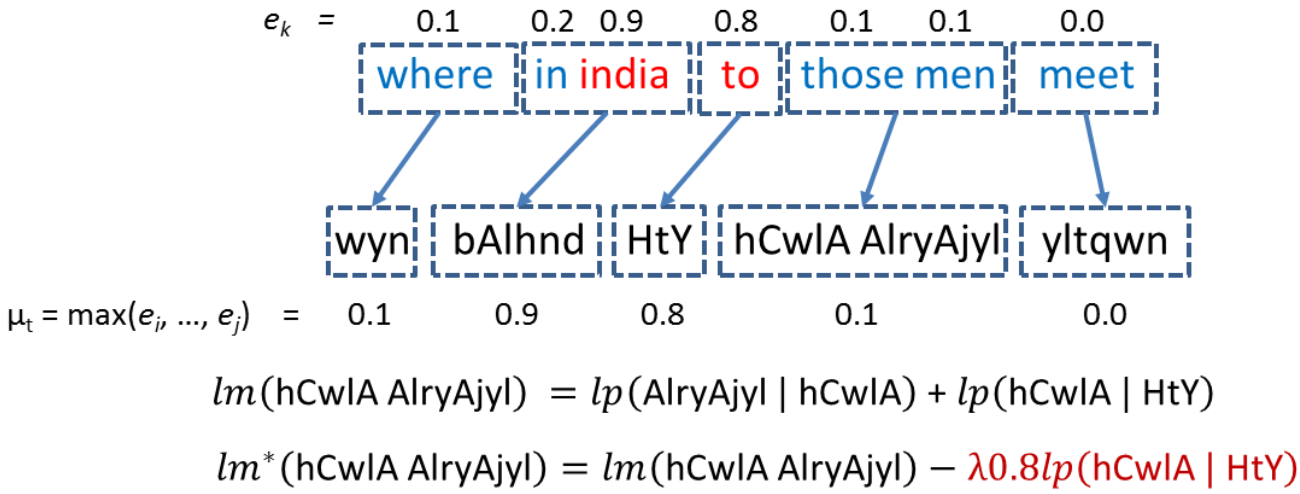


Figure 2: LM penalty highlighted for erroneous bigram context. Incorrect source words are highlighted in red.

beam search stack decoding process. In our decoder, each hypothesis, regardless of which stack it is placed in, records the most recent phrase pair (source/target span) that was used to arrive at that hypothesis. The total LM log-likelihood of the current hypothesis is evaluated as the sum of LM log-likelihoods of each constituent target word given its n-gram context. Depending on the n-gram order, the context may extend to target words from the previous hypothesis on a lower-order stack. If the most recent phrase pair used in obtaining the previous hypothesis corresponds to a source gap, we dynamically adjust LM log-likelihoods for target words in the current hypothesis whose context includes target words from the previous hypothesis.

In the example of Figure 2, the total LM score $lm(\texttt{hCwlA AlryAjyl})$ of the target phrase `hCwlA AlryAjyl` is the sum of the LM log-likelihoods $lp(\texttt{AlryAjyl} \mid \texttt{hCwlA})$ and $lp(\texttt{hCwlA} \mid \texttt{HtY})$ of the constituent words given the local context (without loss of generality, we illustrate using a bigram LM context). However, the bigram likelihood of the first word $lp(\texttt{hCwlA} \mid \texttt{HtY})$ is invalid due to the erroneous context `HtY`, which in turn was generated by the incorrectly recognized English source word (gap) `to`. Therefore, we apply a penalty factor to this term, weighted by the projection of the corresponding source error probability to the target context ($\mu_3 = 0.8$), in addition to the globally tunable feature weight λ. Thus, the penalty term attenuates the effect of the incorrect target hypothesis `HtY` to obtain the discounted LM score $lm^*(\texttt{hCwlA AlryAjyl})$, thereby alleviating the impact of erroneous LM context at run-time.

There is a subtle difference between discounting the total target phrase LM score via subtraction as described above, versus modifying the LM score directly via a multiplicative penalty factor. Our discounting approach is more flexible because it allows us to tune a feature weight specifically for the penalty discount, without affecting the main LM feature weight. In other words, the total penalty can be separated from the total LM score. An alternate solution would have been to back-off to the unigram likelihood of `hCwlA` instead. However, back-off can only be applied with categorical error labels, precluding the use of soft weighting and tunable parameters.

5. Experimental results

To evaluate the proposed approach, we designed and created high-error development/test (HED/HET) sets consisting of spoken utterances rich in OOV entities. Table 1 summarizes these datasets, which exhibit very high OOV/ASR error rates compared to the baseline development/test sets. Consequently, the baseline translation scores of 1-best ASR hypotheses of the HET set were significantly lower (Table 2). For reference, the noise-free test set baseline BLEU and METEOR scores were 16.1 and 42.5, respectively.

We offer a proof-of-concept evaluation of error-aware SMT decoding using oracle source error labels for the HED/HET sets, i.e. with error probability of all correct and incorrect source words set to 0.0 and 1.0, respectively. The oracle error labels were obtained by

Dataset	#Utts	#Words	OOV%	WER
HED	627	6.4k	2.9%	31.8%
HET	507	5.3k	8.9%	46.8%

Table 1: High-error dev/test data statistics.

automatic alignment of ASR hypotheses to the reference transcriptions. Because of the relatively small size of the HED set compared to the baseline development set, we only optimized the weights of the two proposed features on the HED set, carrying over all other tunable parameters from the baseline system. This also allowed a fair comparison, summarized in Table 2, between the baseline and error-aware systems. In combination, the proposed features produced relative gains of 3.2% BLEU and 2.0% METEOR over the baseline system on error-labeled ASR transcriptions of the HET set. Because it is impossible to translate gaps correctly, these improvements are attributable solely to better translations of the islands.

To verify the statistical signficance of this improvement, we performed the non-parametric Wilcoxon signed-rank test based on pair-wise bootstrap resampling [14] of the baseline and error-aware SMT hypotheses. With 100 randomized samples, the p-value returned by this test was 5.14×10^{-17}, thus confirming statistical significance of the improvement at $\alpha = 0.01$.

System	BLEU	METEOR
Baseline	5.67	24.62
EAD (oracle)	**5.85**	**25.12**
EAD (estimated)	5.61	**24.86**

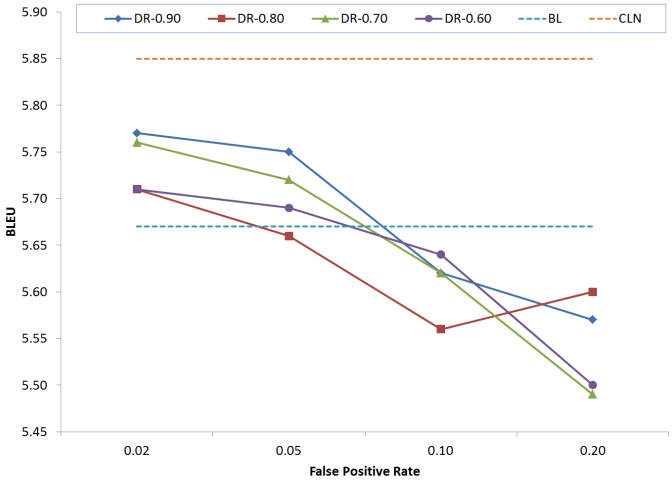
Table 2: HET set translation scores for error-aware decoding (EAD) with oracle/estimated error probabilities.

Achieving perfect ASR error detection is nearly impossible with current technology. We investigated the impact of noisy source error labels on translation performance in order to determine the noise level at which error-aware SMT decoding no longer achieves its goal. We simulated false alarms and missed detections by deliberately injecting noise into the oracle error labels, i.e. randomly changing 0.0 error probabilities to 1.0, and vice-versa in the desired proportion. Figure 3 illustrates the trajectories of BLEU/METEOR scores of error-aware decoding on the HET set across a range of false alarm rates (x-axis). Each curve corresponds to a specific detection rate; for instance, "DR-0.90" refers to 90% error detection rate. Each data point on every curve is the average of 10 independent noise simulations, giving a smooth trajectory of the performance trend. The simulation results are consistent with our intuition that there must be a gradual degradation in translation performance (BLEU/METEOR scores) as the noise level in the source word error labels (false alarm rate) increases. We note that error-aware decoding provides modest BLEU score improvements over the baseline SMT system as long as the false alarm rate is low (2-5%) and detection rate is high (70-80%). METEOR improvements persist at noisier operating points.

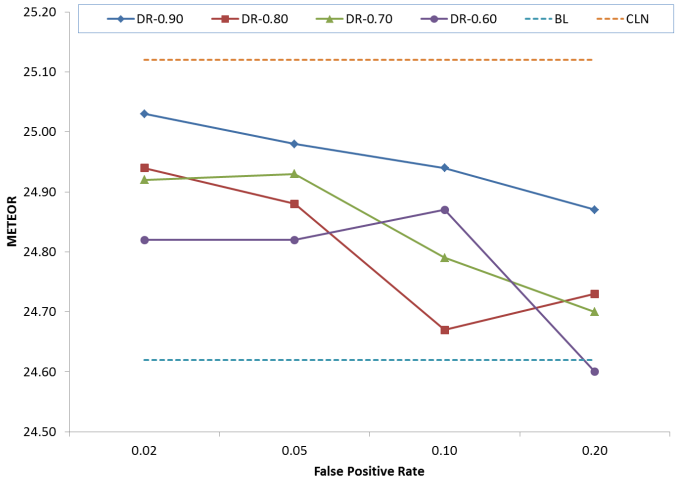
In our final experiment, we attempted to determine whether automatic detection of ASR errors could be used in conjunction with error-aware SMT decoding to improve translation performance in the absence of oracle ASR error labels. To this end, we coupled error-aware SMT decoding with a CRF-based automated ASR error predictor trained on a variety of features, including ASR and SMT confidence scores, subword ASR hypothesis mismatch, word boundary verification, named-entity detection, etc. The predictor infers a real-valued probability of error in [0.0, 1.0] for each source word in the HED/HET sets. Our recent work [15] provides more details on this system. ROC analysis showed that the automated predictor achieved 68% ASR error detection rate at 10% false alarm rate on the HET set. Error probabilities inferred by this system were used to evaluate the proposed penalties for SMT decoding. While the corresponding BLEU score does not improve (final row of Table 2), the METEOR score is slightly better than the baseline system. Given the current performance level of the automated error predictor, these results are in complete agreement with our simulation experiments.

6. Conclusion and future directions

ASR performance is a crucial bottleneck for downstream SMT quality in conversational speech translation systems. Unrecoverable ASR errors due to OOV words can also impact subsequent translation of surrounding, correctly recognized words due to contextual effects. Thus, errors in the source input can cause imperfect or incorrect translation of error-free neighboring words. Besides being less effective on utterances that generate unrecoverable ASR errors, traditional methods of integrating ASR and SMT (for instance, via lattice or



(a) BLEU Trajectories



(b) METEOR Trajectories

Figure 3: Trajectory of BLEU and METEOR scores for error-aware decoding at various false alarm and detection rates for error labels. Dashed horizontal lines represent the baseline (lower) and error-aware decoding with perfect error detection (upper). Figures show a gradual degradation in SMT performance as the noise level in the error labels increases.

n-best based search space augmentation) can be computationally expensive as well as memory intensive.

We presented an exploratory study in which we made targeted modifications to a phrase-based SMT decoder that reduce interference of incorrect gaps on translation of correct islands by introducing dynamic penalties applied to bilingual phrase pairs and the target LM. The new features were directly integrated within the log-linear model, resulting in straightforward development and tuning of the modified SMT system.

In the proof-of-concept experiment where we assumed perfect knowledge of source errors, the proposed modifications gave statistically significant relative improvements of 3.2% BLEU and 2.0% METEOR over the baseline system. Comprehensive simulation experiments revealed that modest translation improvements persist even in the presence of false alarms and missed detections of source errors, subject to certain thresholds. Coupling automated ASR error detection with error-aware SMT decoding yielded small gains in METEOR. We expect translation performance to improve as error prediction accuracy increases.

Based on these observations, one of our primary goals for the future is to improve automated ASR error detection capability for coupling with error-aware decoding. On the other hand, interactive, clarification-enabled SLT systems (e.g. two-way speech-to-speech translation systems) permit us to leverage user feedback to obtain source error labels. For example, based on cues from the automated ASR error detector, the system may request the speaker to confirm whether a sequence of ASR-hypothesized words is incorrect. In this way, user feedback can be used to construct oracle source error labels as input to the error-aware SMT decoder.

7. Acknowledgements

This paper is based upon work supported by the DARPA BOLT program. The views expressed here are those of the author(s) and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

8. References

[1] S. Matsoukas, I. Bulyko, B. Xiang, K. Nguyen, R. Schwartz, and J. Makhoul, "Integrating speech recognition and machine translation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, Honolulu, HI, April 2007, pp. 1281–1284.

[2] Y. Al-Onaizan and L. Mangu, "Arabic ASR and MT integration for GALE," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, Honolulu, HI, April 2007, pp. 1285–1288.

[3] E. Matusov, S. Kanthak, and H. Ney, "Integrating speech recognition and machine translation: Where do we stand?" in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, May 2006, pp. 1217–1220.

[4] R. Zhang and G. Kikui, "Integration of speech recognition and machine translation: Speech recognition word lattice translation," *Speech Communication*, vol. 48, no. 3-4, pp. 321–334, 2006.

[5] L. Mathias, "Statistical machine translation and automatic speech recognition under uncertainty," Ph.D. dissertation, Baltimore, MD, USA, 2008.

[6] R. Kumaran, J. Bilmes, and K. Kirchhoff, "Attention shift decoding for conversational speech recognition," in *INTERSPEECH*, Antwerp, Belgium, August 2007, pp. 1493–1496.

[7] R. Prasad, R. Kumar, S. Ananthakrishnan, W. Chen, S. Hewavitharana, M. Roy, F. Choi, A. Challenner, E. Kan, A. Neelakantan, and P. Natarajan, "Active error detection and resolution for speech-to-speech translation," in *International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, December 2012, pp. 150–157.

[8] L. Nguyen and R. Schwartz, "Efficient 2-pass n-best decoder," in *DARPA Speech Recognition Workshop*, 1997, pp. 167–170.

[9] P. E. Brown, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Computational Linguistics*, pp. 263–311.

[10] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, March 2003. [Online]. Available: http://dx.doi.org/10.1162/089120103321337421

[11] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 48–54.

[12] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ser. ACL '07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 177–180. [Online]. Available: http://dl.acm.org/citation.cfm?id=1557769.1557821

[13] F. J. Och, "Minimum error rate training in statistical machine translation," in *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 160–167.

[14] P. Koehn, "Statistical significance tests for machine translation evaluation," in *EMNLP*, Barcelona, Spain, July 2004, pp. 388–395.

[15] W. Chen, S. Ananthakrishnan, R. Kumar, R. Prasad, and P. Natarajan, "ASR error detection in a conversational spoken language translation system," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 7418–7422.

Evaluation of a Simultaneous Interpretation System and Analysis of Speech Log for User Experience Assessment

Akiko Sakamoto, Kazuhiko Abe, Kazuo Sumita and Satoshi Kamatani

Knowledge Media Laboratory,
Corporate Research & Development Center,
Toshiba Corporation
akiko7.sakamoto@toshiba.co.jp

Abstract

This paper focuses on the user experience (UX) of a simultaneous interpretation system for face-to-face conversation between two users. To assess the UX of the system, we first made a transcript of the speech of users recorded during a task-based evaluation experiment and then analyzed user speech from the viewpoint of UX.

In a task-based evaluation experiment, 44 tasks out of 45 tasks were solved. The solved task ratio was 97.8%. This indicates that the system can effectively provide interpretation to enable users to solve tasks. However, we found that users repeated speech due to errors in automatic speech recognition (ASR) or machine translation (MT). Users repeated clauses 1.8 times on average. Users seemed to repeat themselves until they received a response from their partner users.

In addition, we found that after approximately 3.6 repetitions, users would change their words to avoid errors in ASR or MT and to evoke a response from their partner users.

1. Introduction

This paper focuses on user experience (UX) of our simultaneous interpretation system ([1], Figure 1), which is a variation of a speech-to-speech translation (S2ST) system.

The goal of this paper is to assess whether users are satisfied with the whole conversation process when they use the simultaneous interpretation system and to evaluate whether the system provides interpretation of a quality sufficient for users to obtain information from speakers of other languages.

To assess the UX, we analyzed the transcription of recorded speech during a task-based evaluation experiment. The simultaneous interpretation system consists of several modules: automatic speech recognition (ASR), sentence boundary detection (SBD), machine translation (MT), and user interface (UI). However, from the viewpoint of a user, the whole system is one application. This is why we chose a task-based evaluation experiment when trying to assess UX.

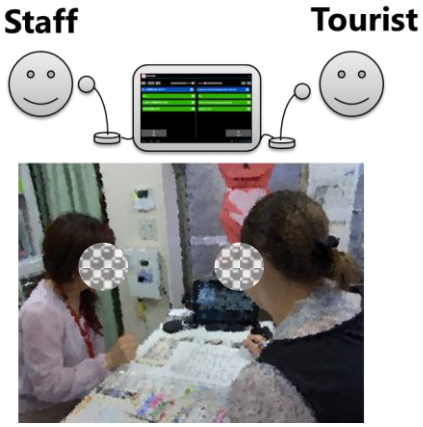


Figure 1*: Our simultaneous interpretation system and users*

Section 2 introduces related work. Section 3 introduces the system that we developed and used for the evaluation experiment. Section 4 describes the evaluation experiment. In section 5, we analyze a transcript of speech recorded during the evaluation experiment and also explore some methods to detect whether users are satisfied with the whole experience of using our system. Section 6 provides a summary of this paper.

2. Related Work

Many studies have targeted S2ST ([2], [3], and [4]). In the early stage of S2ST technology studies, systems were restricted to certain topics and speech styles. Recently, systems that can incrementally interpret utterances have been developed ([5], [6]). Some of them are commercially available [8]. Some complex applications are targeted by S2ST systems, such as lecture interpretation [9].

Most previous studies of S2ST systems have evaluated these systems in terms of recognition, translation accuracy and time efficiency. For example, one simultaneous interpretation system reportedly shortened by 20% the time needed for interpretation

without an accompanying decrease in quality [7].

When developing a simultaneous interpretation system, it is important to evaluate the precision of the interpretation and its time efficiency. In addition, it is important to consider the experience of users during actual use of the system.

Many systems implicitly expect that users will speak rather clearly and fluently. However, those users who are interested in receiving information (e.g., information about shopping), rather than in conversation with the other speaker, do not pay much attention to learning how to use the system. We observed this habit in the conversation of users during task-based evaluation.

Because simultaneous interpretation systems will soon be put to practical use, it is important to pay attention to the UX for the system. It has not been sufficiently discussed what kind of support and UX the system provides. There are few reports on the UX for simultaneous interpretation systems. Here, we focus on the number of repetitions of speech. In the experiment that we discuss in section 4, users repeated similar utterances until the ASR system recognized their speech correctly or until the other speaker responded. We also counted how many times a user would repeat something before changing the spoken words to avoid ASR or MT errors and obtain correct interpretation results and a response from the other user. This means that errors in the ASR or MT system interrupt conversation and decrease user satisfaction.

This paper discusses the UX of the simultaneous interpretation system as measured by repetition of qualitatively identical speech. This paper proposes a guiding principle for developing a practical system of simultaneous interpretation. We developed our own simultaneous interpretation system and evaluated it in terms of conversation goal achievement. We also transcribed speech recorded during the task-based experiment and analyzed how the users spoke.

3. System Architecture

We introduce our simultaneous interpretation system here to clarify the experimental conditions. The simultaneous interpretation system comprises ASR, SBD, MT, and UI components. Figure 2 illustrates the simultaneous interpretation process. The server side engines of the ASR, SBD, and MT components communicate with the UI application, which works as a client terminal through the Internet.

First, the system recognizes the user's spontaneous speech, segmented by 200 ms of pause,

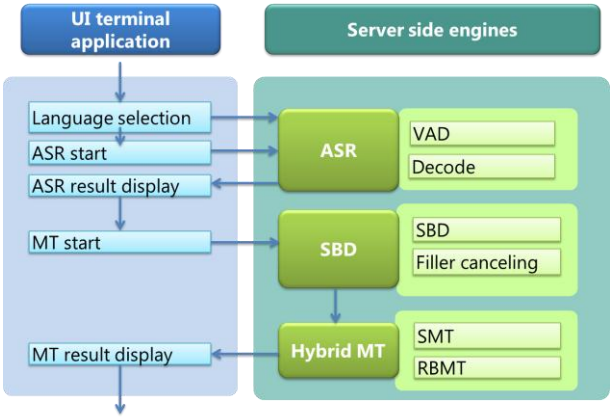


Figure 2*: Schematic diagram of speech production*

Figure 3*: Schematic diagram of speech production*

and then the system continuously outputs a transcribed text. Second, the client terminal UI application gathers several speech segments and sends them to the SBD module. Segments are gathered only when the pause between them are shorter than 500 ms. The SBD module detects a sentence boundary to split the text into segments suitable for translation. Next, the SBD module examines each segment to see whether it needs to be translated. Segments are translated in the order of their speech. This procedure enables the system to start the MT process without waiting for the end of the whole speech by a speaker and to interpret users' utterances after only a short delay for the original user's utterance. In addition, when a user presses a button for text-to-speech (TTS), the TTS engine synthesizes a voice sound for the translation result.

Figure 3 shows an example of the process. The original speech "Excuse me, I lost <pause> a bag at the train station" contains a pause longer than 200 ms between "lost" and "a." Therefore, the ASR engine regards them as separate speech segments of "excuse me i lost" and "a bag at the train station". Next, the

UI application gathers these ASR results and sends them for SBD. The SBD module examines the whole string "excuse me i lost a bag at the train station" and finds a boundary suitable for translation. In the example, SBD found a boundary between "me" and "lost." The system finally outputs the interpretation result for "excuse me" and "i lost a bag at the train station." The rest of this section briefly introduces ASR, SBD, MT and UI, in that order.

3.1. ASR

To achieve accurate speech recognition under noisy environmental conditions, we carefully select the acoustic features for voice activity detection [10] and acoustic modeling [11]. The language model is trained with a large-scale text corpus collected from the web and a bilingual corpus that we developed for the travel domain.

The ASR dictionary contains 200,000 Japanese words and 30,000 English words. These entries are selected according to frequency of appearance in the corpus. In addition, we registered words specific to Kawasaki City in Kanagawa Prefecture, Japan (e.g., names of sightseeing spots, transport facilities, etc.), where we conducted the experiment described in section 4.

We configure the ASR module to output a recognition result for every speech section separated by a 200 ms pause. Because of variety in user speech style, the speech segments processed by ASR are not always appropriate for translation. We introduce an SBD method to provide input text for MT.

3.2. SBD

Among the many works on SBD, [12] is to our knowledge the newest report on SBD for simultaneous interpretation systems. The authors there prepare parallel corpora and create a phrase table using a statistical MT (SMT) tool. They realize SBD by using the phrase table.

In contrast, our SBD is realized by a rather simple process. We first prepared monolingual corpora for Japanese and English. For Japanese, we set sentence boundaries by references to a set of manually developed rules; for English, we regarded punctuation as indicative of boundaries. Next, we used CRF++ [13], a machine-learning tool based on conditional random fields, and created a discrimination process to find sentence boundaries. Through these processes, we obtained monolingual SBD modules for three languages. For Japanese, we added a rule-based filler detector, and sentences that consist of only fillers are deleted as semantically null.

3.2.1. Detection model

Sentence boundaries are detected in two steps. In the first step, the system performs morphological analysis on the results from ASR and obtains word segmentation and also part-of-speech (POS) tags on Japanese and English. Then, fillers and other redundant parts are removed using simple pattern matching to POS.

In the second step, machine-learning-based classifiers detect sentence boundaries. Sentence boundary detection is treated as a labeling task for each word [14]. We prepare spontaneous speech corpus in which words at the beginning of a sentence have "B" labels and other words have "I" labels. We use CRF++ [13] and create a discrimination model for the labeling. For the learning features, we use the surface form of two morphemes before and after each morpheme for Japanese and English.

3.2.2. Training corpus

To create Japanese and English sentence boundary detectors, we used two different corpora: for Japanese, 140,000 sentences from "Corpus of Spoken Japanese (CSJ) [15]", and for English, 110,000 sentences from WIT3 [16] data including transcriptions of TED talks.

These corpora do not contain any tags denoting a suitable unit for translation. We regarded a punctuation mark as a boundary marker in English. For Japanese, we regarded a clause to be a suitable unit for translation [17] and prepared simple rules to find clause boundaries in the training corpus.

3.2.3. Detection performance

We evaluated precision and recall of boundary detection on test sets. The test sets had been ideally segmented into 244 Japanese sentences and 1664 English sentences. We regarded punctuation as definitive segment boundaries. Table 1 shows detection accuracy. In this table, we calculate the precision and recall values as follows:

$$\text{Precision} = \frac{\text{No. of correctly estimated sentence boundaries}}{\text{No. of estimated sentence boundaries}}$$

$$\text{Recall} = \frac{\text{No. of correctly estimated sentence boundaries}}{\text{No. of periods in original corpus}}$$

□

Table 1: *Segment detection accuracy*

	Precision	Recall	F-value
Japanese	0.739	0.672	0.705
English	0.720	0.809	0.763

3.3. MT

3.3.1. Forest-driven rule-based MT

Rule-based machine translation (RBMT) has been used in commercial systems for a long time. A well-developed RBMT engine outputs a better translation and covers a larger domain than other types of systems. However, commercial MT systems are usually designed for use on grammatically written language, and they sometimes fails to process ungrammatically spoken language.

We introduce a forest-driven parsing mechanism ([18], Figure 4) into RBMT. It parses input sentences by generalized LR parsing, which can accept ungrammatical chunks by using an original context-free grammar to capture the clause structure and deal with various ambiguities. The parser then generates possible syntax structures as a forest and transfers the best structure to the target language structure according to syntactic and semantic preferences.

3.3.2. Hybrid MT

SMT can generate natural translation results for restricted and specific domains. RBMT, however, can translate an input sentence robustly, but the result sometimes lacks fluency.

We viewed these strengths and weaknesses as complementary, and so we used SMT and RBMT engines together to form a hybrid MT engine. Specifically, when the probability of an SMT result falls below a specified threshold, the RBMT result is selected instead as the final result of the hybrid MT engine [18]. This engine selection is made for each segment produced by SBD.

We used phrase-based SMT [19]. For Japanese-English and English-Japanese SMT, we trained the engine with a travel domain corpus consisting of 220,000 sentence pairs developed by ourselves and 20,000 sentence pairs distributed by the Advanced Language Information Forum [20].

3.3.3. Translation quality

We evaluated engines both automatically and manually (Table 1). We used the IWSLT 2004 corpus [20] as a test set. For automatic evaluation, 500 sentence pairs were used; the first 100 of these

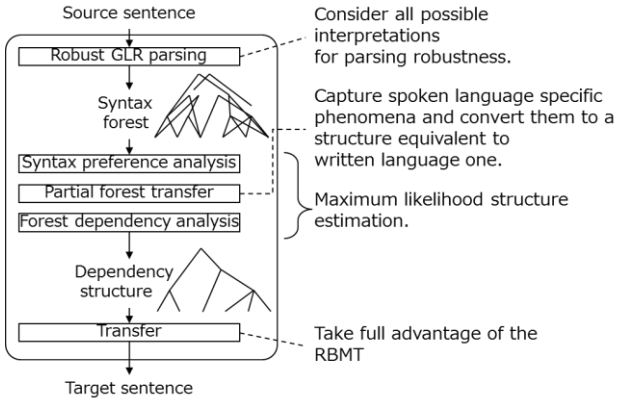


Figure 4*: Process flow of forest driven RBMT*

Table 2: *Detailed Translation Quality (data of IWSLT)*

		Adequacy	Fluency	BLEU	RIBES
E	RBMT	3.93	3.69	20.64	0.575
	SMT	3.90	4.12	33.97	0.650
	Hybrid	4.01	3.89	28.54	0.631
J	RBMT	4.15	3.94	22.21	0.755
	SMT	4.25	4.29	34.28	0.807
	Hybrid	4.30	4.25	32.27	0.790

sentence pairs were used for manual evaluation.

We used BLEU [21] and RIBES [22] for automatic evaluation. We also manually evaluated fluency and adequacy metrics [23]. Table 2 shows the evaluation results. We assumed that adequacy of manual translation reflects correctness of meaning, and we chose the hybrid engine for our simultaneous interpretation system.

3.4. UI

We developed a translation system whose user interface runs on a tablet with the Android operating system. In the task-based assessment, a "host" and a "guest" share a terminal display and communicate with each other through the system.

Figure 5 shows the user interface. A user starts speaking after pressing the "speak" button. While the user continues to speak, it is not necessary to hold the button. When the user presses the button a second time, the system processes it as an explicit signal that speech is concluded.

Until the speech recognition result is finalized, a recognition candidate is shown in gray. When the translation result is finalized, the system displays the ASR and MT text. In Figure 6, the speak button for the English speaker is placed on the right hand side, and the button for the Japanese speaker on the left.

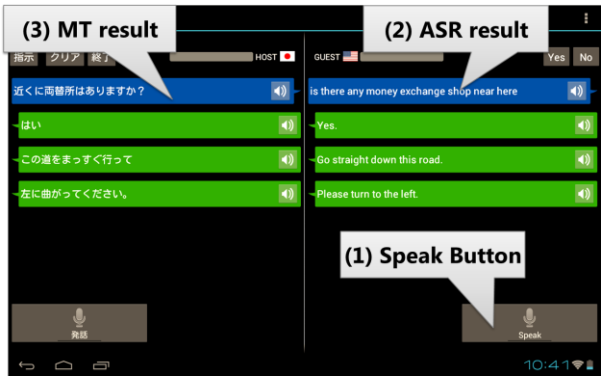


Figure 5: *User interface of Client Application*

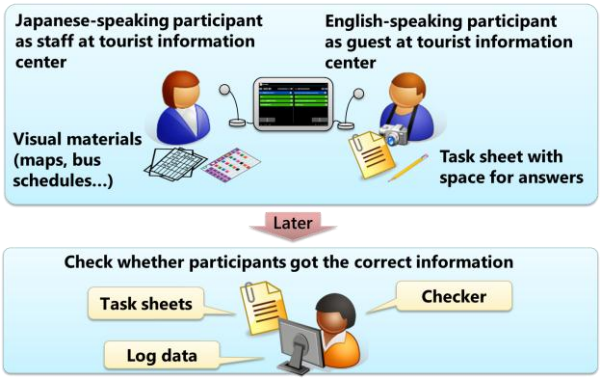


Figure 6: *Experiment situation and the evaluation process of Solved Task Ratio*

For interpretation from English to Japanese, the English speaker presses the speak button (1) and says something, such as "Is there any money exchange shop near here?" After this, the ASR result "is there any money exchange shop near here" is shown on the display (2). Then, the MT result "近くに両替所はありますか [*Chikaku ni ryougaejo wa arimasu ka*]" is shown (3). For Japanese to English, the speak button, ASR result, and MT results are on the opposite side.

4. Task-based Evaluation Experiment

We conducted a task-based evaluation experiment in the Toshiba Customer Service Evaluation Center. This experiment is in addition to a previous evaluation experiment conducted in a tourist information center in Chiba City in Chiba Prefecture, Japan [1]. In this section, we discuss the parts of this prior experiment that relate to the analysis in section 5.

4.1. Tasks

The tasks in the evaluation experiments were as follows. We prepared these tasks on the assumption that the conversation is being held in a tourist information center. The previous experiment [1] was conducted in Chiba City. This additional experiment was held in Kawasaki City in Kanagawa Prefecture. Therefore, we modified some of the tasks to make them appropriate to Kawasaki City. We added 2 tasks to the 8 tasks in [1], and now we have the following 10 travel tasks.

(1) Ask whether you can book any local tours here.
(2) Ask whether you can get to Tokyo Disneyland by train without changing trains.
(3) Ask how much the fare is from Kawasaki Station to Hamamatsucho Station by train.
(4) Ask how to get to a money exchange shop near here.
(5) Now you would like to know the bus route and its schedule in Kawasaki City. Ask how you can get this information.
(6) Ask what is the best souvenir from Japan. Ask about its features and how to get to a store where you can buy it.
(7) Ask your partner to recommend a sightseeing spot and how to get there. Decide whether you will go according to your interest.
(8) Imagine what you would like to try in Japan and ask where you can experience it around here.
(9) Ask how to get downtown from here. Assume that you will have dinner there or go shopping.
(10) You lost your bag on the train. Ask what you should do to find it.

Table 3: *English Speaking Participants*

English Speaking Participant	Sex	Years in Japan	Place of Birth
A	M	3	Los Angels
B	F	3	Hawaii
C	F	3	Arizona
D	M	3	California
E	M	3	South Carolina

Table4: *Japanese Speaking Participants*

Japanese Speaking Participant	Sex	Place of Birth
A	F	Okayama
B	F	Kanagawa
C	F	Tokyo
D	F	Kanagawa
E	M	Tokyo

4.2. Participants and collected data

The data collected for the analysis in section 5 includes conversation logs and transcriptions of five English-speaking participants (Table 3) and of five

Japanese-speaking participants (Table 4). The labels A to E were given to the five pairs of people who had conversations through the system.

4.3. Solved Task Ratio

The solved task ratio indicates the proportion of tasks achieved out of all tasks. In this paper, we focus on 45 tasks for which speech was successfully recorded. Of these, 44 tasks were solved. Therefore, we had a solved task ratio of 97.8%.

5. Analysis of UX

The solved task ratio confirms that our simultaneous interpretation system can almost always help users to obtain information from speakers of a different language. However, we would like to ascertain whether users were satisfied with the whole process of conversation through our system. In other words, we would like to find a way to assess the UX of our simultaneous interpretation system.

5.1. UX for our system

It would be ideal if users would say each thing only once and this speech would be perfectly interpreted by our system. However, since ASR, SBD, and MT do not perform perfectly, users sometimes need to repeat themselves until the partner speaker can understand the interpretation result and respond. It is clear that less frequent repetition is preferable; however, we would still like to determine how many repetitions users will tolerate before experiencing stress. In other words, we would like to know what level of performance is needed so that our system does not put stress on users.

5.2. Statistics from transcript and system log

To assess the UX of the conversation process, we transcribed the 45 conversations from the evaluation experiment and manually analyzed them.
Since spoken language includes parts smaller than clauses, we define here the relationship between "speech," "clause," and "intention of the clause." A "speech" indicates the words from a transcript of the users' voices, terminated by a pause of 200 ms. When spoken slowly, one clause will spread into several speeches, so we manually detected a clause chunk by hand from the transcription. For example, as shown in Figure 7, when a user says, "I want to go," and pauses for 200 ms before saying "on a tour," the speaker uttered two "speeches" but only one "clause." We recorded 1330 speeches during the 45 conversations and manually chunked the speech into

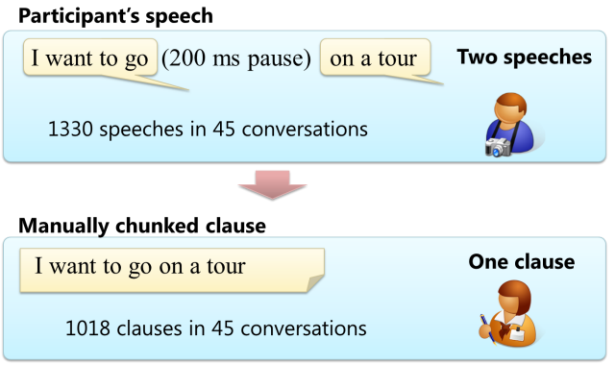


Figure 7*: unit of "a speech sound" and "an utterance"*

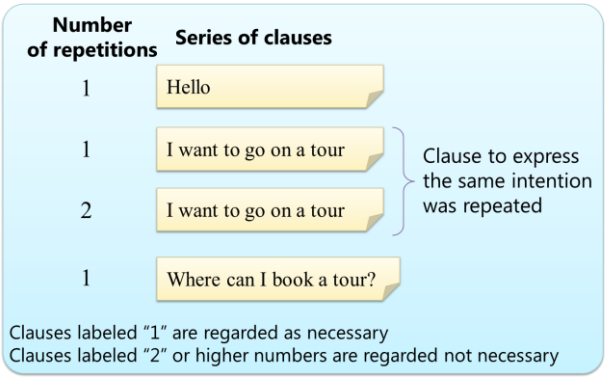


Figure 8*: an example of repeated utterances*

Table 5: *Change of intention after repeated failure of interpretation*

Number of repetition	Transcription of utterances	ASR result
1	Where can I eat Yakiniku?	where can i am eat your key to do it
2	What is a good Yakiniku restaurant?	what is a good jockey to restaurant
-	OK. Where can I get great Sushi?	ok our can i get great sushi

clauses. This gave 1018 clauses in the 45 conversations. The "intention of a clause" indicates the intended meaning of a clause.

5.3. Repeated clauses

We counted how many times clauses were repeated before being understood by the partner speaker. Figure 9 illustrates how we counted the number of repetitions for each clause. In the example, utterances of the same letter are regarded as repetition to express the original intention of the speaker. In this analysis, a question asked by the partner speaker to clarify an unclear interpretation result caused by an interpretation error is also regarded as a repeated utterance.

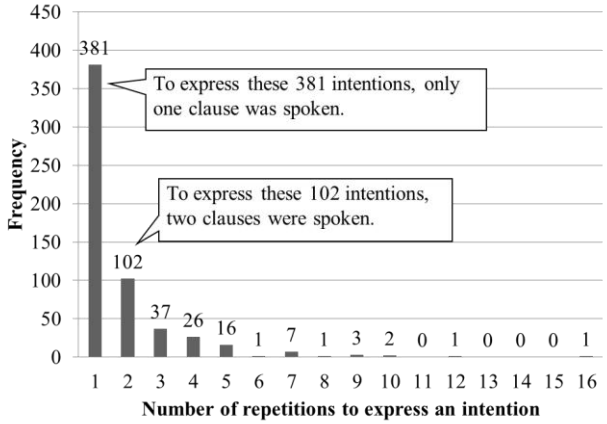


Figure 9: *Number of repeated clauses for 578 intention*

Figure 10 shows the number of intentions that were expressed through multiple, distinct clauses or through more than two repetitions. We found that 381 intentions were expressed through a clause without repetition; 102 intentions were expressed through a clause repeated once. The total number of intentions across the 45 conversations was 578.

To assess whether the number of repetitions was too large, we used another measure. As shown in Table 5, the speaker originally wished to eat "yakiniku," which is a Japanese-style grilled meat. However, the word "yakiniku" was not recognized well and so was not interpreted to get the response from the partner speaker. The speaker changed to asking about "sushi" instead; this was successfully recognized and interpreted, and the partner speaker responded. The speaker did not return to the original intention of "yakiniku" again. In this example, an ASR error caused the interpretation error, but in some other cases, the ASR succeeded and MT caused an interpretation error.

In the 45 conversations, there were 6 intentions that were changed due to repeated utterances. The speaker changed intentions after an average of 3.6 interpretation errors (as indicated by lack of response from the partner speaker).

6. Conclusions

We introduced our simultaneous interpretation system for face-to-face conversation between two people, and we also analyzed the transcription of the speech and the system log in the experiment. This new version of our system has a revised SBD module. In the new system, several speeches are first combined together and then the system finds a suitable unit for translation.

We also evaluated the system by a task-based experiment. The evaluation experiment showed a solved task ratio of 97.8% across 45 tasked-based conversations. However, we found that users repeated each utterance 1.8 times on average.

From analysis of the transcripts and the system log, we found that after approximately 3.6 interpretation errors, users would change what they said to avoid interpretation error and receive a response from the partner user. For future work, we would like to improve our system to reduce user speech repetition.

7. References

[1] A. Sakamoto et al., "Development of a Simultaneous Interpretation System for Face-to-Face Services and Its Evaluation Experiment in Real Situation," In *Proc. Machine Translation Summit XIV*, Nice, France, 2013, pp.85-92.

[2] A. Waibel et al., "JANUS: a speech-to-speech translation system using connectionist and symbolic processing strategies," In *Proc. ICASSP'91*, Toronto, 1991, pp.793-796.

[3] F. Metze et al., "The NESPOLE! speech-to-speech translation system," In *Proc. HLT 2002*, San Diego, CA, 2002.

[4] W.Wahlster, "Verbmobil: translation of face-to-face dialogs," In *Proc. 3rd European Conf. on Speech Communication and Technology*, Berlin, 1993, pp.29-38.

[5] S. Matsubara and Y. Inagaki, "Incremental Transfer in English-Japanese Machine Translation," *IEICE TRANSACTIONS on Information and Systems*, Vol.E80-D, No.11, pp.1122-1130, 1997.

[6] S. Bangalore et al., "Real-time Incremental Speech-to-Speech Translation of Dialogs," In *Proc. NAACL-HLT 2012*, Motreal, 2012, pp.437-445.

[7] H. Shimizu et al., "Constructing an Automatic Simultaneous Interpretation System using Simultaneous Interpretation Data," In *Proc. The 2013 Autumn Meeting of the Acoustic Society of Japan*, Toyohashi, 2013, pp.59-62.

[8] NTT docomo, 2012, *NTT DOCOMO to Introduce Mobile Translation of Conversations and Signage*, Available: http://www.nttdocomo.co.jp/english/info/media center/pr/2012/001611.html

[9] C. Fügen , A. Waibel, M. Kolss, "Simultaneous translation of lectures and speeches," Machine Translation, 21, pp.209-252, (2007).

[10] H. Ding et al., "Comparative evaluation of different methods for voice activity detection,"

In *Proc. Interspeech 2008*, Brisbane, 2008, pp.107-110.
[11] M. Nakamura et al., "Evaluation of Group Delay based Features in Noisy Environments," In *Proc. The 2012 Spring Meeting of the Acoustic Society of Japan*, 2012, Yokohama, pp.947-952.
[12] G. Neubig et al., "A method for deciding translation timing in speech translation considering reordering between languages," In *Proc. The 2013 Autumn Meeting of the Acoustic Society of Japan*, Toyohashi, 2013, pp.55-58.
[13] Y. Liu et al., "Using Conditional Random Fields For Sentence Boundary Detection In Speech," In *Proc. of the 43rd Annu. Meeting of ACL*, Ann Arbor, MI, pp.451-458, (2005).
[14] T. Kudo, 2005, *CRF++: Yet Another CRF toolkit*, Available: https://code.google.com/p/crfpp/
[15] K. Maekawa et al., "Spontaneous Speech Corpus of Japanese," In *Proc. LREC 2000*, Athens, 2000, pp.947-952.
[16] M. Cettolo et al., "WIT3: Web inventory of transcribed and translated talks," In *Proc. EAMT 2012*, Trento, 2012, pp.261-268.
[17] K. Takanashi et al., "Identification of "Sentence" in Spontaneous Japanese – Detection and modification of clause boundaries –," In *Proc. SSPR 2003*, Tokyo, 2003, pp.183-186.
[18] S. Kamatani et al., "Hybrid Spoken Language Translation Using Sentence Splitting Based on Syntax Structure," In *Proc. Machine Translation Summit XII*, Ottawa, 2009.
[19] H. Wang et al., "The TCH Machine Translation System for IWSLT 2008," In *Proc. IWSLT 2008*, Waikiki, HI, 2008, pp.124–131.
[20] Y. Akiba et al., "Overview of the IWSLT04 evaluation campaign," In *Proc. IWSLT 2004*, Kyoto, 2004, pp.1-12.
[21] K. Papineni et al., "BLEU: a method for automatic evaluation of machine translation," In *Proc. the 41st Annu. Meeting of ACL*, Sapporo, 2002, pp.311-318.
[22] H. Isozaki et al., "Automatic Evaluation of Translation Quality for Distant Language Pairs," In *Proc. EMNLP 2010*, Cambridge, MA, 2010, pp.944-952.
[23] P. Koehn and C. Monz, "Manual and automatic evaluation of machine translation between European languages," In *Proc. the HTL-NAACL Workshop on Statistical Machine Translation*, New York, NY, 2006, pp.102-121.

Parameter Optimization for Iterative Confusion Network Decoding in Weather-Domain Speech Recognition

Shahab Jalalvand, Daniele Falavigna

Human Language Technology unit, Fondazione Bruno Kessler, via Sommarive 18, Trento, Italy
{jalalvand,falavi}@fbk.eu

Abstract

In this paper, we apply a set of approaches to, efficiently, rescore the output of the automatic speech recognition over weather-domain data. Since the in-domain data is usually insufficient for training an accurate language model (LM) we utilize an automatic selection method to extract domain-related sentences from a general text resource. Then, an N-gram language model is trained on this set. We exploit this LM, along with a pre-trained acoustic model for recognition of the development and test instances. The recognizer generates a confusion network (CN) for each instance. Afterwards, we make use of the recurrent neural network language model (RNNLM), trained on the in-domain data, in order to iteratively rescore the CNs. Rescoring the CNs, in this way, requires estimating the weights of the RNNLM, N-gramLM and acoustic model scores. Weights optimization is the critical part of this work, whereby, we propose using the minimum error rate training (MERT) algorithm along with a novel N-best list extraction method. The experiments are done over weather forecast domain data that has been provided in the framework of EUBRIDGE project.

Key words: automatic speech recognition, language model, neural network, confusion network, minimum error rate training

1. Introduction

A major problem in domain-specific speech recognition is the lack of sufficient in-domain data for acoustic modeling and language modeling. In the case of language modeling, one could train a n-gram based LM using a huge set of out of domain data and, then, adapt it to the domain using a given set of in-domain data and some adaptation techniques such as the ones described in [1], [2] and [3].

In this paper, we focus on the language modeling part and we introduce efficient approaches for post-processing the output of the automatic speech recognition (ASR) system. The recognizer generates the word graphs for each utterance. Then, we convert them into the Confusion Network (CN) forms. This form yields better oracle word error rate (WER) in comparison to the N-best list and word graphs. Then, we go through an iterative decoding approach for rescoring the confusion networks.

For rescoring the CNs, we adopt an approach similar to the one described by A. Deoras [4], in particular we combine, using iterative decoding, word posterior, RNNLM and NgramLM probabilities. The RNNLM is trained on the small (about 1 million words) set of in-domain data, which consists of captioning of weather forecast news. The reason for using RNNLM is that it has proven to exhibit good performance even if trained on small sizes of training data [5]. In order to estimate the weights to be assigned to RNNLM, NgramLM and posterior probability scores, we utilize Minimum Error Rate Training (MERT) technique [6] along with a novel method for extracting the N-best lists from the CNs.

In Section 2 we will describe the acoustic models and the baseline LM employed in the experiments, as well as the process for generating word graphs and confusion networks. A description of the iterative decoding approach is given in Section 4. In Section 4 we describe the MERT approach developed for learning the weights of the various models used in the rescoring step. Section 5 describes the development/test corpora used and reports the experiments and results. Finally, Section 6 concludes the paper.

1.1. Related Works

The N-gram language model is commonly used in speech recognition systems. Simplicity and low computational complexity are the most important factors of this type of language model which has made it quite popular among the researchers. During the years, different extensions have been made on top of this model to overcome its deficiencies such as data sparseness, generalization and curse of dimensionality. The back-off techniques [7] and the discounting methods [8], [9] are the main extensions over the N-gram LM which are mostly based on making an interpolation between the shorter contexts. However, since in the N-gram LM the words are seen as discrete entities, computing interpolation between their probabilities is, in principle, not possible.

An attempt to change the representation of the words in language modeling was done by Y. Bengio [10], when he introduced the neural network LM. In this model, the words are represented as the binary vectors. Schwenck [11] added a projection layer to the NNLM and named it the continuous space language model. The projection layer converts the binary word vectors into the real number vectors. He also applied this model in a large vocabulary continuous speech recognition system. The probability of the words in these feed forward NNLMs depends on a limited context (the same as the N-gram LMs). T. Mikolov [5] proposed the recurrent neural network LM in which the context is not constrained by a Markov window. The recursive arcs in the hidden layers work as a cache to save the impact of the previous words.

These neural network approaches have shown better performance in terms of Perplexity; however, applying them

directly in the ASR decoder is costly in computation and memory. A common solution is to utilize these models for rescoring the N-best list produced by a traditional ASR decoder which uses a finite state network constructed from a lexicon and an N-gram LM.

However, N-best list rescoring is not the best way to benefit from the high potential of the NNLMs, as the number of the hypotheses limited. For example in our case, the oracle word error rate of the 1000-best list is around 9.9%, while, the word error rate of the 1-best is 10.4%. One could see that there is no big gap in-between. Instead of the N-best list, it is also possible to rescore word graphs or confusion networks. In our case, the oracle word error rate of the word graphs and the confusion networks resulted to be 5.5% and 3.4%, respectively.

2. ASR training and CN generation

For training acoustic models (AMs) we have used audio data provided within the EUBRIDGE consortium containing recordings of weather forecasts. These recordings come with captioning which is not exact transcriptions of the audio so that, in order to train tri-phone Hidden Markov Models (HMMs) a preliminary alignment step is carried out between automatic transcriptions of the training data and the corresponding given captioning. Hence, only the segments of audio recordings that align with the corresponding captioning are retained for HMM training. After this phase about 30 hours of the weather forecasts have been selected for AM training.

For language modeling, we are given a set of weather forecast sentences consisting of about 1 million words. With this latter set of sentences we train an in-domain LM which, in turn, is used for automatically selecting from a large general corpus (see [18]), containing about 1.6 billions of words, the sentences with the lowest perplexity. The automatically selected sentences, formed by about one hundred million words, are used to train a 4-gram, back-off LM which is finally adapted, using the "mix" adaptation method described in [2] to the in-domain data.

From the 4-gram adapted LM, we generate a finite state network (FSN), which also embeds the lexicon, that is used in two ASR decoding passes (the details of the ASR decoder are given in [14]).

Word graphs (WGs) are generated in the second decoding pass. To do this, all of the word hypotheses that survive inside the trellis during the Viterbi beam search are saved in a word lattice containing the following information: initial word state in the trellis, final word state in the trellis, related time instants and word log-likelihood. From this data structure and given the LM used in the recognition steps, WGs are built with separate acoustic likelihood and LM probabilities associated to the word transitions. To increase the recombination of paths inside the trellis and consequently the density of the WGs, the so called word pair approximation [16] is applied. In this way the resulting graph error rate was estimated to be around 33% of the corresponding WER.

Consensus decoding, through confusion network (CN) generation, allows minimizing the word error rate (WER) of sentence hypotheses, instead of maximizing the related posterior probability or, equivalently, minimizing the sentence error rate [15]. A CN is formed by a concatenation of confusion bins, each containing a list of word hypotheses with related posterior probabilities. Basically, a CN is generated from a given WG by: 1) identify CN bins inside the WG corresponding to the non-overlapped time windows, 2) merge all the transitions inside a bin that share the same word (word posterior in a bin is the sum of all the corresponding link posterior in the original WG). In this work, the CNs are produced using the algorithm described in [15] and the software package described in [17].

3. Iterative CN decoding

The method of iterative Confusion Network decoding has already been proposed by A. Deoras [4]. Thus, for further details, we refer the readers to this paper. Here, we briefly describe this method with some variations in our own work.

As mentioned above, a confusion network is a concatenation of bins. The process of iterative decoding, starts from the first bin, re-orders the arcs and shifts to the next one. In each bin, the decoder generates some hypotheses. The number of these hypotheses is equal to the number of the arcs in that bin. Different hypotheses are created by changing a word in the sentence with the words of the bin. Thus, all the hypotheses in each bin differ in just one word. To each hypothesis, the feature functions assign a score. The feature functions, in our case, are RNNLM, N-gramLM, Posterior and Length (the number of the words). The lengths of the hypotheses may differ if there is a null arc in the bin. Then, the scores are interpolated and the resulted score is used to re-order the arcs. After finishing processing a bin, the decoder moves to the next bin and repeats this step. By reaching at the last bin, the score of the best hypothesis (the one which is obtained by concatenating the first arcs) is computed. If this score is better than the one obtained from the previous iteration, the decoder continues this step, otherwise, it stops.

To illustrate the process, we assume a confusion network (*CN*) consisting of four bins (*A, B, C* and *D*):

$$CN : \{A[a_1..a_{n_a}], B[b_1..b_{n_b}], C[c_1..c_{n_c}], D[d_1..d_{n_d}]\}$$

Here, n_a is the number of the arcs in the bin *A*, and so on. Each bin contains a number of arcs and some contents which are assigned to the arcs. These contents are: a word, a posteriori score, an LM score and an acoustic model score. Thus, each arc can be seen as a structure:

$$\begin{cases} a_i.w \rightarrow a\ word \\ a_i.p \rightarrow a\ posteriori\ score \\ a_i.lm \rightarrow a\ language\ model\ score \\ a_i.am \rightarrow an\ acoustic\ model\ score \end{cases}$$

The arcs in each bin are ordered according to their posteriori scores. Hence, the 1-best hypothesis (e^*) in CN is made by concatenating the first arcs:

$$e^* = a_1.w, b_1.w, c_1.w, d_1.w$$

$a_1.w$ is the word assigned to the first arc of the bin *A*. When the decoder starts processing the first bin (A), it will generate n_a different hypotheses:

$$e = \begin{Bmatrix} e_1 = a_1.w, b_1.w, c_1.w, d_1.w \\ e_2 = a_2.w, b_1.w, c_1.w, d_1.w \\ \dots \\ e_{n_a} = a_{n_a}.w, b_1.w, c_1.w, d_1.w \end{Bmatrix}$$

Note that the hypotheses are different in just one word. In order to compare them, we need to compute the new scores.

The RNNLM and NgramLM scores can be computed by applying the LMs on this set of sentences. For the posteriori scores, we can sum up the posteriors of all the words in each sentence or just consider the posteriori of the changing words. Finally, the total score of a sentence is computed by ($i=1..n_a$):

$$\begin{aligned} score(e_i) = \; & \lambda_{rnnlm} \times rnnlm(e_i) + \\ & \lambda_{ngram} \times ngramlm(e_i) + \\ & \lambda_{poster} \times posteriori(e_i) + \\ & \lambda_{length} \times length(e_i) \end{aligned} \quad (1)$$

The length function should be taken into account to avoid being biased towards the short/long sentences. The weights (λ) can be estimated on a development set and by using the optimization techniques.

The critical parts of this method are: selection of the feature functions, and estimation of the weights. In the next section, we describe the MERT algorithm which is a type of machine learning approach for estimating the weights.

4. Minimum Error Rate Training

The MERT algorithm was first introduced by F. Och [6] for using in a statistical machine translation (SMT) task. The algorithm is based on training a parameter model on a set of N-best targets and optimizing the model. The optimized model generates a new set of N-best targets. This set is merged with the one from the previous iteration.

For a reference instane like f_s we aim at finding a candidate in e (that is the corresponding N-best list) which maximizes the total score.

$$\hat{e}(f_s;\lambda_1^M) = \arg\max_{e \in C_s} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e \mid f_s) \right\} \quad (2)$$

In the equation, C_s is the N-best list suggested for f_s. The parameters h_m and λ_m are the function and weight of the m^{th} feature, respectively. In our case, we have four feature functions: RNNLM, N-gramLM, Posterior and length.

The optimized weights for the feature functions can be obtained by solving a minimization problem over the error function $E(r_s,e_s)$.

$$\hat{\lambda}_1^M = \arg\min_{\lambda_1^M} \left\{ \sum_{s=1}^{S} E(r_s, \hat{e}(f_s;\lambda_1^M)) \right\} \quad (3)$$

The value S is equal to the number of the sentences in the development set.

In the extended version of MERT developed by N. Bertoldi et al. [12], the algorithm is run in two loops: the outer loop and the inner loop. Starting from initial weights in the outer loop, the decoder processes the input instances and generates the corresponding N-best list. This list is used to feed the inner loop where the weights are optimized. The inner loop continues optimizing the weights till the time that there is no big change in the weights.

The new weights are again used to run the decoder and generate the new N-best lists. In order to make sure that there is enough diversity among the N-best lists, the new list is combined with the previous one. The outer loop is iterated until the time that no considerable change is observed in WER.

4.1. The M-best Extraction Method

The decoder that is used in our work has been explained in the Section 2.1. The output of this decoder is an N-best list which is extracted from the confusion network. Given a confusion network, one could use a simple A* search algorithm to extract the N-best list from the network. This method that is already embedded in SRI toolkit uses the posterior scores of the arcs in order to output the N-bests. Since, the value of N is limited, the number of the hypotheses will be limited. Therefore, there would be some words in some bins that can never be seen among the hypotheses. It means that, the rescoring process might be again entangled in the lack of hypotheses. This is exactly the problem that is existed with simply rescoring the N-best lists.

In this paper, we propose an efficient method for extracting the candidate list for MERT and we call it "M-best list". In this method, all the possible hypotheses that can be generated in each bin are merged and considered as the N-best list of that step. Therefore, assuming CN as the decoded confusion network, the extracted M-best list includes:

$$e = \begin{Bmatrix} e_1 = a_1.w, b_1.w, c_1.w, d_1.w \\ \dots \\ e_i = a_{n_a}.w, b_1.w, c_1.w, d_1.w \\ e_{i+1} = a_1.w, b_2.w, c_1.w, d_1.w \\ \dots \\ e_{i+1} = a_1.w, b_{n_b}.w, c_1.w, d_1.w \\ \dots \\ e_{i+2} = a_1.w, b_1.w, c_{n_c}.w, d_1.w \\ \dots \\ e_M = a_1.w, b_1.w, c_1.w, d_{n_d}.w \end{Bmatrix}$$

Note that the maximum size of M would be equal to:

$$n_a + (n_b - 1) + (n_c - 1) + (n_d - 1)$$

While, the maximum number of the hypotheses is:

$$n_a \times n_b \times n_c \times n_d$$

The advantages of this method are: 1) the MERT algorithm can see and process all the possible words in its inner loop; 2) there is no boundary for the size of M. According to the size of the confusion network, the number of the sentences could be different, while in the traditional method, this size is limited to N.

The scores of each of these sentences are computed as before. The posterior score of a sentence is also computed by summing up all the posteriors of the words in the sentences.

5. Experiments and Results

In this section, we first describe the details of the corpus that is exploited in this work. Then, we go through the experiments. The reported experiments are arranged as follows:

- Generating the confusion networks on the development and test instances.
- Using Grid search approach for estimating the weights
- Using MERT approach for estimating the weights

We perform these experiments on two sets of confusion networks: one generated using the Bi-gramLM and the other generated using the 4-gramLM.

5.1. The Corpus

The dataset that we have used to analyze and evaluate our approaches is in the domain of weather forecast news,

provided for the EU-BRIDGE project. As mentioned in the Section 2, in this dataset there is an in-domain text set that is around 1 Million words. This data has been used to train the RNNLM and also to select the auxiliary data from the out-of-domain resource. There is also a domain-related text set about 100 MW that has been selected automatically (see Section 2 for the method of selection). The latter set is used to train the Bi-gramLM and 4-gramLM that are used along with the pre-trained acoustic models to generate the ASR output and also the Confusion Networks.

The development and test sets contain 32 and 650 utterances, respectively. The MERT algorithm is run over the development set, in order to estimate and optimize the desired weights for rescoring. Obtaining the optimized weights, the iterative decoding is performed on the test set to rescore the confusion networks.

5.2. Experiments

By using the IRSTLM toolkit [13], we train a Bi-gram and a 4-gram back-off, modified shift beta smoothed language models on the domain-related set (100MW) and we used them in the ASR decoder for generating two different sets of word graphs (one with Bi-gram and one with 4-gram LM). The ASR engine, used for this task is described in [14]. Afterwards, we use the SRI toolkit [17] to convert the word graphs into the confusion networks. At the end, we have two different sets of confusion networks: one created by using the Bi-gramLM and the other by 4-gramLM. The motivation of generating these two sets is to assess the performance of the iterative decoding approach (by the Bi-gram CNs), and improving the results (by the 4-gram CNs).

The confusion networks created in this way contain lots of useless bins with null arcs. This number of useless bins dramatically increases the computational cost. Hence, we filter the confusion networks according to the posterior of the null arcs, i.e. all the bins containing null arcs with higher posterior than 0.99 are eliminated. This filtering decreases the average number of the bins per CN up to 92 percent (without changing the WER).

The resulted CNs yield 16.4% and 10.4% WER on the development set and 20.2% and 14.3% WER on the test set for both Bi-gram and 4-gram CN sets, respectively (see Table 1 and 2).

In order to rescore the confusion networks, we use a RNNLM trained on the in-domain data. The RNNLM is built by the toolkit developed by T. Mikolov, et al [5]. For combining the scores from RNNLM, 4-GramLM, posterior and length, a simple linear interpolation is applied. In order to estimate the weights of these feature functions, we chase two different methods: Grid search and MERT.

For applying the Grid search algorithm, we simply consider an interval from zero to one to assign a weight to each feature function:

$$\begin{cases} \lambda : \{\lambda_{rnnlm}, \lambda_{ngramlm}, \lambda_{posterior}, \lambda_{length} \in [0:0.1:1]\} \\ s.t.\ \lambda_{rnnlm} + \lambda_{ngramlm} + \lambda_{posterior} + \lambda_{length} = 1 \end{cases} \quad (4)$$

By each set of the values, we decode the development confusion networks and the best set is selected to be used on the test set. One could find the results of this method in the second row of the Tables 1 and 2.

Furthermore, we use the MERT algorithm on the development set. In this way, we exploit the proposed method for extracting the M-best lists at the end of each iteration of the decoder. Then, MERT is run to process the M-best list and optimize the weights. On this development set, MERT usually stops at the fourth of fifth iteration. A reason could be the lack of the feature functions. Here, we have just four functions that might not be sufficient. Another reason is the lack of the development data. Nevertheless, in order to validate the weights, suggested by MERT, we ran the algorithm several times on the development set and we selected the best one. The results of this method can be found in the third row of the Tables 1 and 2.

Table 1: The WER results on the confusion networks created by the Bi-gramLM

	Dev	Test
Baseline	16.4	20.2
RNNLM-Grid-ItDec	14.1	18.9
RNNLM-MERT-ItDec	13.5	18.3

Table 2: The WER results on the confusion networks created by the 4-gramLM

	Dev	Test
Baseline	10.4	14.3
RNNLM-Grid-ItDec	10.2	14.3
RNNLM-MERT-ItDec	9.5	14.0

As it can be seen from the tables, the results of the confusion networks created by using the 4-gramLM are apparently better, because the 4-gramLM is more accurate. Note, that the training set and the procedure of training these two LMs are completely the same. Exactly because of the same reason, the improvement in the experiment on the Bi-gramLM is higher. Again, note that the RNNLM used for rescoring both sets of confusion networks is the same. Therefore, one could evaluate the performances of the iterative decoding and the MERT algorithm. Finally, we can see a slight improvement by using the MERT algorithm over the Grid search. It means that the weights suggested by MERT are more efficient than the Grid search. Moreover, the number of iterations taken by MERT is fewer. For example, in MERT, the weights are estimated in 4 or 5 iterations, while for Grid search, we need 66 iterations (according to the intervals considered for the weights in Eq. 4).

There are some deficiencies in the experiments:

- The size of the development set is small and insufficient to have a better weight estimation.
- There are a few feature functions that are not enough for MERT to give a reliable estimation.
- The size of the training set of the RNNLM (1MW) is not comparable with the N-gramLM (100MW).

Considering these deficiencies, we are designing the future experiments, in particular by using more RNNLMs. Due to the complexity of the RNNLM structure, it's not efficient to build it on the big training sets. A wise solution would be to train several RNNLMs on the separated parts of the training set, and then use them as the new feature functions.

6. Conclusion

A set of approaches were introduced and analyzed for improving the process of rescoring the domain-specific ASR output. Instead of the common N-best list rescoring, we used

confusion network rescoring that yields better oracle WER. An iterative decoding approach was used for rescoring the confusion networks and improving the output. Additionally, we applied the MERT algorithm to optimize the weights of the feature functions more efficiently. We also introduced a novel approach for extracting the N-best list from the confusion network that improves the affect of MERT optimization process.

7. Acknowledgement

This work has been partially founded by the European project EU-BRIDGE, under the contract FP7-287658

8. References

[1] Federico, M. (1999, September). Efficient language model adaptation through MDI estimation. *In Proceedings of Eurospeech*.

[2] Foster, G., & Kuhn, R. (2007). Mixture-model adaptation for SMT. *In Proceedings of the Second Workshop on Statistical Machine Translation* (pp. 128-135), Prague, Czech Republic

[3] Ruiz, N., Federico, M., & Kessler, F. F. B. (2012). MDI Adaptation for the Lazy: Avoiding Normalization in LM Adaptation for Lecture Translation. *In Proceedings IWSLT 2012*.

[4] Deoras, A., & Jelinek, F. (2009, November). Iterative decoding: A novel rescoring framework for confusion networks. *In proceedings of Automatic Speech Recognition & Understanding (ASRU 2009)* (pp. 282-286).

[5] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. *In proceedings of INTERSPEECH (*pp. 1045-1048).

[6] Och, F. J. (2003, July). Minimum error rate training in statistical machine translation. *In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1* (pp. 160-167).

[7] Katz, S. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *35*(3), (pp. 400-401).

[8] Witten, I. H., & Bell, T. C. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, *37*(4) (pp. 1085-1094).

[9] Chen, S. F., & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *In Computer Speech & Language*, *13*(4) (pp. 359-393).

[10] Bengio, Y., Schwenk, H., Senécal, J. S., Morin, F., & Gauvain, J. L. (2006). Neural probabilistic language models. *In Innovations in Machine Learning* (pp. 137-186), Springer Berlin Heidelberg.

[11] Schwenk, H. (2007). Continuous space language models. *In Computer Speech & Language*, *21*(3), (pp. 492-518).

[12] Bertoldi, N., Haddow, B., & Fouet, J. B. (2009). Improved minimum error rate training in Moses. *In The Prague Bulletin of Mathematical Linguistics*, *91*(1) (pp. 7-16).

[13] Federico, M., Bertoldi, N., & Cettolo, M. (2008, September). IRSTLM: an open source toolkit for handling large scale language models. *In proceedings of INTERSPEECH* (pp. 1618-1621).

[14] Falavigna, D., Gretter, R., Brugnara, F., & Giuliani, D. (2012, December). FBK@ IWSLT 2012-ASR track. *In Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*.

[15] Mangu, L., Brill, E., & Stolcke, A. (2000). Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *In Computer Speech & Language*, *14*(4) (pp. 373-400).

[16] Ney, H., Ortmanns, S., & Lindam, I. (1997, April). Extensions to the word graph method for large vocabulary continuous speech recognition. *IEEE International Conference on Acoustics, Speech, and Signal, ICASSP-97* (Vol. 3, pp. 1791-1794).

[17] Stolcke, A. (2002, September). SRILM-an extensible language modeling toolkit. *In proceedings of INTERSPEECH*.

[18] Falavigna, D., & Gretter, R. (2012). Focusing Language Models for Automatic Speech Recognition. *In Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*.