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 produced 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

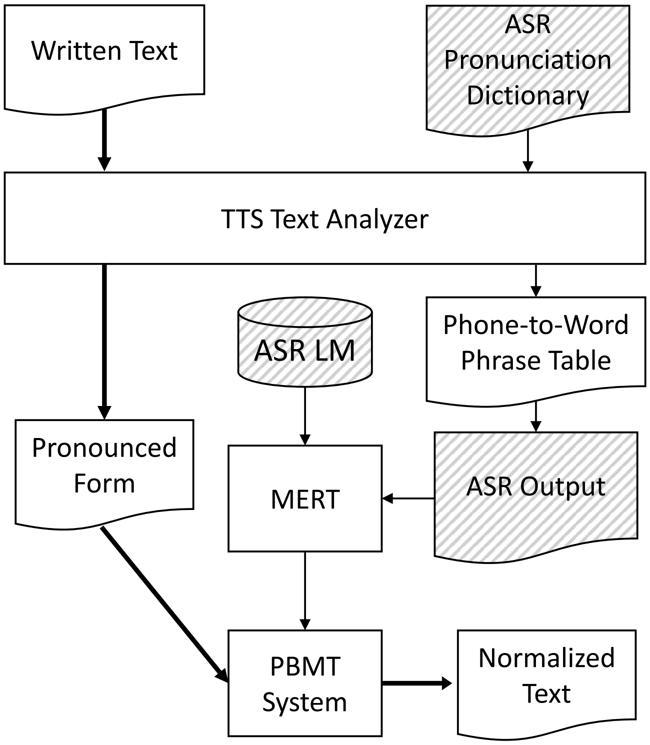

Figure 1: Phonetic normalization pipeline

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

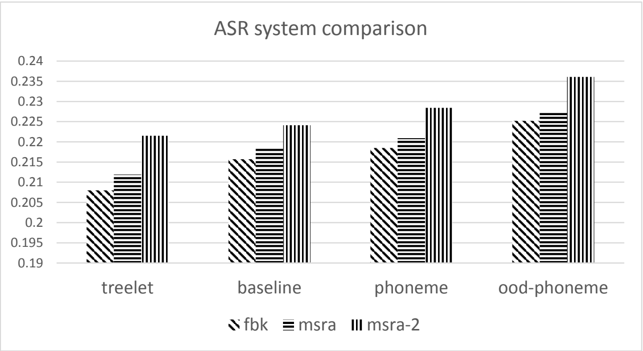

Figure 2: BLEU Results by ASR system

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.

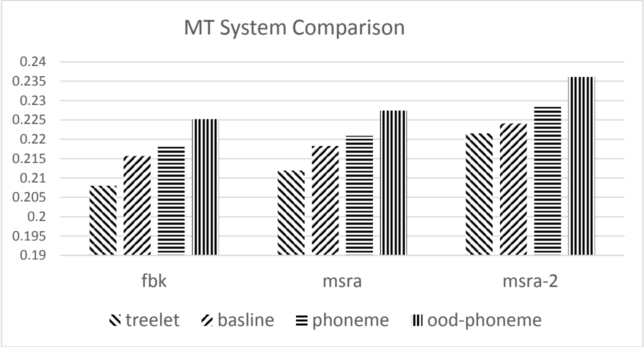

Figure 3: BLEU Results by MT system

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) 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.