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 $i^{\text{th}}$ 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 $\ell_2$ 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.