The RWTH Aachen Machine Translation System for IWSLT 2016

Jan-Thorsten Peter, Andreas Guta, Nick Rossenbach, Miguel Graça, 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 of *International Workshop on Spoken Language Translation* (IWSLT) 2016. We have participated in the MT track for the German→English language pair employing our state-of-the-art phrase-based system, neural machine translation implementation and our joint translation and reordering decoder. Furthermore, we have applied feed-forward and recurrent neural language and translation models for reranking. The attention-based approach has been used for reranking the *n*-best lists for both phrase-based and hierarchical setups. On top of these systems, we make use of system combination to enhance the translation quality by combining individually trained systems.

1. Introduction

We describe the statistical machine translation (SMT) systems developed by RWTH Aachen University for the evaluation campaign of IWSLT 2016. We participated in the machine translation (MT) track in the Talk and MSLT task for the German→English language pair. The combination of multiple machine translation systems has proven to be highly effective on this task. The individual engines include state-of-the-art neural machine translation (NMT), phrase-based translation (PBT) and the joint translation and reordering (JTR) systems. Each of these is a sound system including preprocessing, translation and postprocessing. As a final step, we combine all of these systems using our system combination implementation.

The preprocessing used in this work is described in Section 2. In Section 3 we describe the translation software and setups we have made use of. It is split up into the description of our NMT (Section 3.1), PBT (Section 3.3) and JTR system (Section 3.3). Section 3.4 describes the applied language models and Section 3.5 the long short term memory (LSTM) language and translation models used in rescoring. Section 3.6 explains the system combination pipeline applied on top of the individual systems. Our experiments for each track are summarized in Section 4 and we conclude with Section 5.

2. Preprocessing

In this section we briefly describe our preprocessing pipeline, which is equal to our IWSLT 2015 German→English preprocessing pipeline [1].

2.1. Categorization

We worked on the categorization of the digits and written numbers for the translation task. All written numbers were categorized. As the training data and also the test sets contain several errors for numbers in the source as well as in the target part, we put effort into producing correct English numbers. In addition, ',' and '.' marks were inverted in most cases, as in German the former mark is the decimal mark and the latter is the thousand separator.

2.2. Compound Splitting and POS-based Word Reordering

We reduced the source vocabulary size for the German→English translation and split the German compound words with the frequency-based method described in [2]. To reduce translation complexity, we employed the long-range part-of-speech based reordering rules proposed by [3]. In this regard, we did no further morphological analysis in our preprocessing pipeline.

3. SMT Systems

For the IWSLT 2016 evaluation campaign, state-of-the-art neural machine translation, phrase-based and joint translation and reordering systems have been utilized. GIZA++ [4] is employed to train word alignments. The systems are evaluated case-sensitive on the BLEU [5], TER [6] and CharacTER [7] measures. The `TED.dev2010` and `TEDX.dev2012` development sets are used for optimization. Based on preliminary results, our impression is that the `TED.dev2010` set is closer related to the TED talks and the `TEDX.dev2012` to the MSLT task [8].

3.1. Neural Machine Translation System

A main component of our provided system is a neural machine translation system (NMT) similar to [9]. We use an implementation based on Blocks [10] and Theano [11, 12].

The decoder and encoder word embeddings are of size 620, the encoder uses a bidirectional layer with 1000 LSTMs [13] to encode the source side. A layer with 1000 LSTMs is used by the decoder. The data is converted into subword units using byte pair encoding with 20000 operations [14]. During training a batch size of 50 is used. The applied gradient algorithm is AdaDelta [15].

We train multiple neural networks with slightly different settings and use them to create four ensembles. The differences between the networks are two different kinds of preprocessing, one as described in Section 2 and the other is equal to our IWSLT 2013 system [16]. The main reason for combining NNs based on two different preprocessings is the performance benefit of complementary translation models in system combination. In addition, we use various methods to provide the alignment computation with supplementary information, linguistic coverage, word fertility, context dependency, context gating and guided alignment [17, 18, 19]. Some networks are trained with a dropout of 20%, where others are trained for a few additional iterations on the TED or QED data, respectively.

Two ensembles have been created using the preprocessing described in Section 2 and all available bilingual data, while the remaining two use the IWSLT 2013 system [16]. In both cases, the first ensemble is created by choosing the networks and the number of training iterations based on its performance measured in BLEU on the `TED.dev2010` set, in order to perform well on the TED talk task. On the other hand, the second ensemble is created based on the networks' performance on the `TEDX.dev2012` set, which seems to be more similar to the the MSLT task. In total, this results in four different ensembles.

3.2. Phrase-based System

Our phrase-based decoder (PBT) is the implementation of the *source cardinality synchronous search* (SCSS) procedure described in [20] in RWTH's open-source SMT toolkit, Jane 2.3[1] [21], which is freely available for non-commercial use. 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, n-gram target language models and enhanced low frequency feature [22]. The parameter weights are optimized with MERT [23] towards the BLEU metric. Additionally, we make use of a hierarchical reordering model (HRM) [24], a high-order word class language model (wcLM) [25], a joint translation and reordering (JTR) model (cf. Section 3.3), whose integration is described in [1], and reranking using neural network models (cf. Sections 3.1 and 3.5).

[1] `http://www-i6.informatik.rwth-aachen.de/jane/`

3.3. Joint Translation and Reordering System

This system combines the flexibility of word-level models with the search accuracy of phrase candidates. It incorporates the joint translation and reordering (JTR) model [26], a language model (LM) and two lexical models for smoothing purposes. Phrases annotated with word alignments are utilized in SCSS decoding to hypothesize many-to-many translation candidates.

3.3.1. Training and Marginalization

A JTR sequence $(\tilde{f},\tilde{e})_1^{\tilde{I}}$ is an interpretation of a bilingual sentence pair and the word alignment. Thus, the probability $p(f_1^J,e_1^I,b_1^I)$ can be estimated as the joint probability $p((\tilde{f},\tilde{e})_1^{\tilde{I}})$. We abbreviate the history $(\tilde{f},\tilde{e})_{i-n+1}^{i-1}$ by h_i:

$$p(f_1^J,e_1^I,b_1^I) = p((\tilde{f},\tilde{e})_1^{\tilde{I}}) = \prod_{i=1}^{\tilde{I}} p((\tilde{f},\tilde{e})_i|h_i). \qquad (1)$$

The Viterbi alignments are obtained using GIZA++ and converted together with the bilingual sentence pairs into JTR sequences. The JTR model $p((\tilde{f},\tilde{e})_i|h_i)$ is estimated with interpolated modified Kneser-Ney smoothing [27] using the KenLM toolkit [28]. We extend the *joint* model by JTR *conditional* models for both translation directions: $p(\tilde{f}_i|\tilde{e}_i,h_i)$ and $p(\tilde{e}_i|\tilde{f}_i,h_i)$. The source conditional probability is computed from the joint probability:

$$p(\tilde{f}_i|\tilde{e}_i,h_i) = \frac{p((\tilde{f},\tilde{e})_i|h_i)}{\sum_{\tilde{f}} p((\tilde{f},\tilde{e}_i)|h_i)}. \qquad (2)$$

In order to compute the target marginal probability $\sum_{\tilde{f}} p((\tilde{f},\tilde{e}_i)|h_i)$, its corresponding ARPA file is generated by processing the ARPA file for the joint probability iteratively for all m-grams for $m = 1,\ldots,n$. The source marginals are generated analogously.

3.3.2. Translation Candidate Extraction

It is crucial to provide the phrases with word alignment annotations, as the models applied in search operate on the level of words. We extract all many-to-many phrases that are consistent with the word alignments from the bilingual sentence pairs using the refined method proposed in [29].

In order to have translation candidates that are more adaptive to the source sentence, we follow the approach of [30] and concatenate up to three continuous candidates during extraction. The resulting discontinuous candidates allow to skip up to two sequences of source words. Target sequences have to be continuous.

3.3.3. Log-Linear Features

The general domain and in-domain 5-gram JTR joint models [26] are responsible for estimating the translation and reordering probabilities in mutual context. Additionally, general domain JTR conditional models are included into the

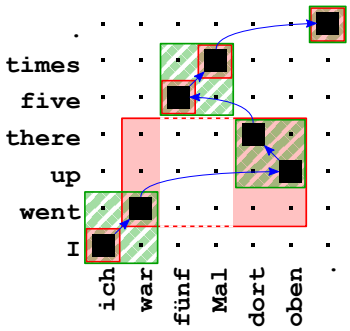

Figure 1: An example of multiple phrasal segmentations taken from the common crawl corpus. The JTR sequence is indicated by blue arcs. The distinct phrasal segmentations are shown in red and shaded green colour.

log-linear framework. To avoid a high overlap with the JTR joint model, they include no context beyond phrasal boundaries. The scores are computed offline and stored in the translation table, which also includes the relative frequencies of the translation candidates. The LMs are estimated as n-gram models with interpolated modified Kneser-Ney smoothing, as described in Section 3.4. Because the JTR models are trained on Viterbi aligned word-pairs, they are limited to the context provided by aligned word pairs as well as sensitive to the quality of the word alignments. To overcome this issue, we incorporate IBM 1 lexical models for both directions. A deep bidirectional translation model is applied in rescoring 1000-best lists for the system optimized on the dev2010 corpus, see Section 3.5.

The heuristic features used by the decoder are an enhanced low frequency penalty [22], a word bonus, a penalty for unaligned source words and a symmetric word-level distortion penalty. Thus, different phrasal segmentations have the same reordering costs if they are equal in their word alignments. The decoder also incorporates gap, open gap and gap distance penalties [31]. All parameter weights are optimized using MERT [23] towards the BLEU metric.

3.3.4. Decoding

Search is performed synchronously to the source cardinality [20]. For each hypothesis expansion, the corresponding word alignment is retrieved from the translation candidates table and used to generate the JTR sequence and to compute the lexical, reordering and gap penalty scores. The JTR model and the LM scores are computed using KenLM. The corresponding histories are stored in the search states. The last aligned source position of each hypothesis expansion is also stored, since it is needed for the computation of future reordering and gap costs as well as the JTR reordering token. States are recombined when they are equal in the source coverage, last aligned source position, LM and JTR model histories. As illustrated in Figure 1, several segmentations of the same sentence pair can be equal in their word alignments.

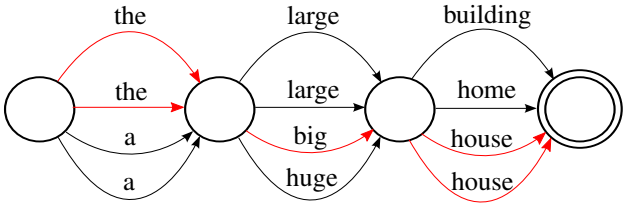

Figure 2: System A: *the large building*; System B: *the large home*; System C: *a big house*; System D: *a huge house*; Reference: *the big house*.

During decoding, this is an issue, as most of the search graph would be filled with numerous hypothesis expansions that share equal word alignments and translations. Therefore, if two states are to be recombined, we first check whether they are equal in their JTR sequences and accordingly delete one of them to avoid duplicates.

3.4. Backoff Language Models

The PBT and JTR systems use three backoff language models that are estimated with the KenLM toolkit [28] and integrated into the decoder as additional features in the log-linear combination. They include a large general domain 5-gram LM, an in-domain 5-gram LM and a 7-gram word class language model (wcLM). All of them use interpolated modified Kneser-Ney smoothing. For the general domain LM, we first select $\frac{1}{2}$ of the English Shuffled News, French Shuffled News and both the English and French Gigaword corpora by the cross-entropy difference criterion described in [32]. The selection is then concatenated with all available remaining monolingual data and used to build the language model. The in-domain language model is estimated on the TED data only. For the word class LM, we train 200 classes on the target side of the bilingual training data using an in-house tool similar to mkcls. With these class definitions, we apply the technique shown in [25] to compute the wcLM on the same data as the general-domain LM.

3.5. Recurrent Neural Network Models

The PBT system applies reranking on 1000-best lists using a recurrent LM. The recurrency is handled with the long short-term memory (LSTM) architecture [33] and we use a class-factored output layer for an increased efficiency as described in [34]. In addition, we apply a deep bidirectional word-based translation model (RNN-BTM) described in [35]. The neural networks are trained using 2000 word classes and equivalent to the general-domain language models used in the IWSLT 2014 and IWSLT 2015 evaluations [36, 1]. The in-domain bilingual data is used for training the RNN-BTM. We use 200 nodes in the forward and backward projection layers, the first hidden layers for both forward and backward processing and the second hidden layer, which joins the output of the directional hidden layers. The neural networks

Table 1: Results of the individual systems for the German→English MT task.

#	System	Opt.	TED.tst2010			TED.tst2014			TEDX.tst.2014			MSLT.dev2016		
			BLEU	TER	CTER	BLEU	TER	CTER	BLEU	TER	CTER	BLEU	TER	CTER
1	NMT 2013 5best	TED	34.3	45.0	43.4	32.3	48.4	47.6	25.2	56.9	55.3	36.9	43.9	39.6
2	NMT 2016 8best	TED	34.6	44.7	42.8	33.7	47.4	46.7	24.7	59.3	54.9	39.0	41.9	36.2
3	NMT 2013 5best	TEDX	34.2	44.7	43.4	32.3	47.9	47.7	25.7	56.0	55.1	37.9	42.4	39.2
4	NMT 2016 8best	TEDX	33.4	44.9	43.6	32.6	47.1	47.5	26.4	55.4	54.7	40.8	39.3	34.8
5	PBT	TEDX	30.5	49.0	45.5	29.4	51.6	49.9	25.2	56.5	54.1	38.6	39.9	35.3
6	PBT + JTR	TEDX	31.7	47.1	45.1	30.4	50.1	49.4	26.3	54.8	55.9	39.8	38.5	34.9
7	PBT + LSTM LM + NMT	TEDX	31.8	47.2	44.6	30.8	49.6	48.4	27.1	53.9	52.9	41.6	36.4	32.2
8	JTR	TEDX	31.8	46.8	45.9	30.6	49.7	49.5	26.0	54.0	56.7	38.9	38.7	36.5
9	PBT + JTR + NMT	TED	33.1	46.7	44.1	32.1	49.6	48.0	25.9	56.1	54.1	39.9	40.2	36.4
10	JTR + LSTM BTM	TED	32.2	47.5	45.4	30.8	50.3	49.5	24.6	56.8	55.7	37.6	40.7	37.6
11	JTR + LSTM BTM (updated)	TED	32.2	47.3	45.3	30.9	50.1	49.5	24.9	56.2	55.5	37.8	40.4	37.3
12	NMT syscomb 1-4	-	34.3	44.4	42.7	33.4	47.1	46.7	26.2	56.4	54.1	40.3	40.8	36.9
13	TED syscomb 1-4, 5, 7, 8	-	35.0	44.1	42.7	34.2	46.5	46.9	27.6	53.1	55.6	42.9	37.6	36.9
14	MSLT syscomb 1-4, 9, 10	-	34.7	44.1	42.9	33.8	46.7	46.9	27.9	53.2	54.3	43.0	37.6	35.4

The NMT ensembles are depending on the mark selected to perform good on the TED or MSLT task and use either the 2013 or 2016 preprocessing (pp). The TED syscomb (#13) is our final system for the TED talk task and the MSLT syscomb (#14) for the MSLT task. The updated JTR system (#11) is submitted as a contrastive system, as it performs better than the previously optimized JTR system (#10) which is part of our system combination (#14). Optimization TED means that the `TED.dev2010` set was used to optimize the model weights using MERT for PBT and JTR or to select the models for the NMT ensembles. TEDX means `TEDX.dev2012` was used. CTER stands for CharacTER.

were implemented using the RWTHLM toolkit[2].

3.6. System Combination

System combination is applied to produce consensus translations from multiple hypotheses which are obtained from different translation approaches. The consensus translations outperform the individual hypotheses in terms of translation quality. A system combination implementation developed at RWTH Aachen University [37] is used to combine the outputs of different engines.

The first step in system combination is the generation of confusion networks (CN) from I input translation hypotheses. We need pairwise alignments between the input hypotheses. The alignments are obtained by METEOR [38]. The hypotheses are then reordered to match a selected skeleton hypothesis regarding the order of words. We generate I different CNs, each having one of the input systems as the skeleton hypothesis. The final lattice is the union of all I-many generated CNs. Figure 2 depicts an example of a confusion network with $I = 4$ input translations. The decoding of a confusion network is finding the shortest path in the network. Each arc is assigned a score of a linear model combination of M different models, which include a word penalty, a 3-gram LM trained on the input hypotheses, a binary primary system feature that marks the primary hypothesis and a binary voting feature for each system. The binary voting feature for a system outputs 1 if the decoded word origins from that system and 0 otherwise. The different model weights for the system combination are trained with MERT.

[2] `https://www-i6.informatik.rwth-aachen.de/web/Software/rwthlm.php`

Table 2: Comparison to last years German→English MT task submission.

System	TED test 2010			TEDX test 2014		
	BLEU	TER	CTER	BLEU	TER	CTER
2015-Submission	31.9	47.6	45.5	26.2	54.7	54.6
TED-system	35.0	44.1	42.7	27.6	53.1	55.6
MSLT-system	34.7	44.1	42.9	27.9	53.2	54.3

4. Experimental Evaluation

The performance of the individual MT systems is summarized in Table 1. The NMT ensembles show a strong performance, especially on the TED data sets. The best NMT ensemble outperforms the best PBT system by 1.5 BLEU on `TED.tst2010`. On the `TEDX` data sets, our strongest PBT system was able to beat the strongest NMT ensemble by 0.8 BLEU on `MSLT.dev2016`.

All NMT systems profited from combining multiple networks into ensembles, the strongest single network for `TED.tst2010` scored 1.9 BLEU worse then the best ensemble. For `MSLT.dev2016` the improvement of using an ensemble was 1.7 BLEU.

The performance of the JTR decoder (#8) is on par with the PBT system that also includes the JTR model (#6) on the TED and TEDX sets, but applying NMT in rescoring on top of the PBT system outperforms the JTR decoder. Including the JTR system into the system combinations provided valuable additional information, showing that count-based and neural models complement each other.

Combining the different ensembles using the system combination did not show improvements over the strongest single NMT system, but combining NMT, PBT, and JTR gave an overall improvement of 0.4 BLEU for the TED.tst2010 and 1.4 BLEU for the MSLT.dev2016 sets. This underlines that system combination work best when applied on systems that significantly differ from each other.

It is worth noting that even though the BLEU score improves for all system combinations, sometimes the TER and CharacTER do not. On the MSLT.dev2016 test set, the TER score drops by 1.2 and the CharacTER score drops by 3.2 points, system #7 to system combination #14. Improvements in TER can often be explained by shorter translations: For the single system (#7), the average translation length is 97.7% of the reference length, whereas the system combination (# 14) output has an average length of 99.2% in comparison to the reference.

5. Conclusion

In comparison to last year's submission, we have moved from using phrase-based and hierarchical systems towards NMT systems combined with PBT and JTR systems. Last year's system combination is improved by 3.1 BLEU for the TED.tst2010 set and by 1.7 BLEU on the TEDX.tst2014 set as shown in Table 2.

Our best performing single system for the TED task is an ensemble of 8 NMT networks. For the MSLT task, we achieve our best results without system combination by using a PBT system with a LSTM language model and a NMT system in rescoring. Using a system combination of multiple different systems results in a significant boost of 1.4 BLEU for the TED and MSLT task on top of the best single system.

6. Acknowledgements

This paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n^{o} 645452 (QT21).

7. References

[1] J.-T. Peter, F. Toutounchi, S. Peitz, P. Bahar, A. Guta, and H. Ney, "The rwth aachen german to english mt system for iwslt 2015," in *International Workshop on Spoken Language Translation*, Da Nang, Vietnam, Dec. 2015, pp. 15–22.

[2] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *Proceedings of European Chapter of the ACL (EACL 2003)*, 2003, pp. 187–194.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] W. Wang, J.-T. Peter, H. Rosendahl, and H. Ney, "Character: Translation edit rate on character level," in *ACL 2016 First Conference on Machine Translation*, Berlin, Germany, Aug. 2016.

[8] C. Federmann and W. D. Lewis, "Microsoft speech language translation (mslt) corpus: The iwslt 2016 release for english, french and german," in *International Workshop on Spoken Language Translation*, Seattle, USA, Dec. 2016.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," May 2015.

[10] B. van Merriënboer, D. Bahdanau, V. Dumoulin, D. Serdyuk, D. Warde-Farley, J. Chorowski, and Y. Bengio, "Blocks and fuel: Frameworks for deep learning," *CoRR*, vol. abs/1506.00619, 2015. [Online]. Available: http://arxiv.org/abs/1506.00619

[11] 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.

[12] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[14] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," pp. 1715–1725, August 2016.

[15] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[16] 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 *International Workshop on Spoken Language Translation*, Heidelberg, Germany, Dec. 2013, pp. 88–93.

[17] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Coverage-based neural machine translation," *CoRR*, vol. abs/1601.04811, 2016. [Online]. Available: http://arxiv.org/abs/1601.04811

[18] T. Cohn, C. D. V. Hoang, E. Vymolova, K. Yao, C. Dyer, and G. Haffari, "Incorporating structural alignment biases into an attentional neural translation model," *CoRR*, vol. abs/1601.01085, 2016. [Online]. Available: http://arxiv.org/abs/1601.01085

[19] W. Chen, E. Matusov, S. Khadivi, and J. Peter, "Guided alignment training for topic-aware neural machine translation," *CoRR*, vol. abs/1607.01628, 2016. [Online]. Available: http://arxiv.org/abs/1607.01628

[20] 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.

[21] 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, pp. 483–491.

[22] B. Chen, R. Kuhn, G. Foster, and H. Johnson, "Unpacking and transforming feature functions: New ways to smooth phrase tables," in *MT Summit XIII*, Xiamen, China, Sept. 2011, pp. 269–275.

[23] 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.

[24] 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*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 848–856.

[25] 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.

[26] A. Guta, T. Alkhouli, J.-T. Peter, J. Wuebker, and H. Ney, "A Comparison between Count and Neural Network Models Based on Joint Translation and Reordering Sequences," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015.

[27] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," Computer Science Group, Harvard University, Cambridge, MA, Tech. Rep. TR-10-98, Aug. 1998.

[28] 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*, Sofia, Bulgaria, August 2013, pp. 690–696.

[29] F. J. Och and H. Ney, "The Alignment Template Approach to Statistical Machine Translation," *Computational Linguistics*, vol. 30, no. 4, pp. 417–449, Dec. 2004.

[30] M. Huck, E. Scharwächter, and H. Ney, "Source-side discontinuous phrases for machine translation: A comparative study on phrase extraction and search," *The Prague Bulletin of Mathematical Linguistics*, no. 99, pp. 17–38, Apr. 2013.

[31] 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

[32] R. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *ACL (Short Papers)*, Uppsala, Sweden, July 2010, pp. 220–224.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Interspeech*, Portland, OR, USA, Sept. 2012.

[35] M. Sundermeyer, T. Alkhouli, J. Wuebker, and H. Ney, "Translation modeling with bidirectional recurrent neural networks," in *Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, Oct. 2014, pp. 14–25.

[36] J. Wuebker, S. Peitz, A. Guta, and H. Ney, "The rwth aachen machine translation systems for iwslt 2014," in *International Workshop on Spoken Language Translation*, Lake Tahoe, CA, USA, Dec. 2014.

[37] M. Freitag, M. Huck, and H. Ney, "Jane: Open source machine translation system combination," in *Proc. of the Conf. of the European Chapter of the Assoc. for Computational Linguistics (EACL)*, Gothenberg, Sweden, Apr. 2014, pp. 29–32.

[38] 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, June 2005, pp. 65–72.