The Karlsruhe Institute of Technology Systems for the News Translation Task in WMT 2016

†Thanh-Le Ha, †Eunah Cho, †Jan Niehues, †Mohammed Mediani,
†Matthias Sperber, *Alexandre Allauzen and †Alexandre Waibel
†Karlsruhe Institute of Technology, Karlsruhe, Germany
*LIMSI-CNRS, Orsay, France
†firstname.surname@kit.edu *surname@limsi.fr

Abstract

In this paper, we present the KIT translation systems as well as the KIT-LIMSI systems for the ACL 2016 First Conference on Machine Translation. We participated in the shared task Machine Translation of News and submitted translation systems for three different directions: English→German, German→English and English→Romanian.

We used a phrase-based machine translation system and investigated several models to rescore the system. We used neural network language and translation models. Using these models, we could improve the translation performance in all language pairs we participated.

1 Introduction

Following the research we have been conducted over previous years, in this paper, we describe our phase-based translation systems submitted to the First Conference on Machine Translation with the highlights on our new models.

In this evaluation, we mainly focused on using neural models in rescoring of a phrase-based machine translation system. We used three different types of neural models: a factored neural model, the continuous space translation models developed by LIMSI and a recurrent encoder-decoder model.

The paper is organized as follows: the next section gives a detailed description of our systems including the hightlighted models. The translation results for all directions are presented afterwards and we then close with a conclusion.

2 System Description

In this section, we first describe our common models we used in our baseline systems. Then specific models and new methods applied in this evaluation will be described.

2.1 Baseline Systems

For training our systems, we used all the data provided by the organizers.

In all of our translation systems, the preprocessing step was conducted prior to training. For English→Romanian, we used the preprocessing described in (Allauzen et al., 2016). For the systems involving German and English, it includes removing very long sentences and the sentence pairs which are length-mismatched, normalizing special symbols and smart-casing the first word of each sentence. In the direction of German→English, compound splitting (Koehn and Knight, 2003) was applied on the German side of the corpus. To improve the quality of the Common Crawl corpus being used in training, we filtered out noisy sentence pairs using an SVM classifier as described in (Mediani et al., 2011).

All of our translation systems are basically phrase-based. An in-house phrase-based decoder (Vogel, 2003) was used to generate all translation candidates from the word lattice and then the weights for the models were optimized following the Minimum Error Rate Training (MERT) method (Venugopal et al., 2005).

The word alignments were produced from the parallel corpora using the GIZA++ Toolkit (Och and Ney, 2003) for both directions. Afterwards, the alignments were combined using the *grow-diag-final-and* heuristic to form the phrase table. It was done by running the phrase extraction scripts from Moses toolkit (Koehn et al., 2007).

Unless stated otherwise, we used 4-gram language models (LM) with modified Kneser-Ney smoothing, trained with the SRILM toolkit (Stolcke, 2002). In the decoding phase, the LMs were scored by KenLM toolkit (Heafield, 2011). In

Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers, pages 303–310,
Berlin, Germany, August 11-12, 2016.

addition to word-based language models, we employed various types of non-word language models in our translation systems. They included bilingual LMs, cluster LMs and the LMs based on POS sequences. For cluster and POS-based LMs, we used an n-gram size of nine tokens. During decoding, these language models were used as additional models in the log-linear combination.

A family of lexical translation models, which we called discriminative word lexicon (DWL), were also utilized in our translation systems. A discriminative word lexicon, first introduced by (Mauser et al., 2009), is a lexical translation model which calculates the probability of a target word given the words of the source sentence. (Niehues and Waibel, 2013) proposed an extension of DWL where they use n consecutive source words as one feature, thus they could incorporate better the order information of the source sentences into classification. In addition to this DWL, we integrated a DWL in the reverse direction in rescoring. We will refer to this model as source DWL (SDWL). This model predicts the target word for a given source word using numbers of context features as described in details in (Herrmann et al., 2015).

To deal with the differences in word order between source and target languages, our systems employed various reordering strategies, which are described in the next section.

2.2 Reordering Models

In all translation directions, the reordering models based on POS tags were applied to change the word positions of the source sentence according to the target word order. In order to train such reordering models, probabilistic rules were extracted automatically from the POS-tagged training corpus and the alignments. The rules cover short-range reorderings (Rottmann and Vogel, 2007) as well as long-range reorderings (Niehues and Kolss, 2009). The POS tags were generated using the TreeTagger (Schmid and Laws, 2008).

Besides the POS-based reordering models, a tree-based reordering model, as described in (Herrmann et al., 2013), was also applied to better address the differences in sentence structure between German and English in our systems. We used the Stanford Parser (Rafferty and Manning, 2008; Klein and Manning, 2003) to generate syntactic parse trees for the source sentences in the training data. Then the tree-based reordering rules were learnt based on the word alignments between source and target sentences, showing how to reorder the source constituents to match the word order of the corresponding target side.

The POS-based and tree-based reordering rules were applied to each input sentence to generate all reordered variants of the sentence. Then a word lattice was produced, encoding the original sentence order as well as those variants. The lattice was then used as the input to the decoder.

In addition, we utilized a lexicalized reordering model (Koehn et al., 2005), which encodes possible reordering orientations (monotone, swap or discontinuous) of each word and its original position in the phrase pair. Hence, it can be learnt directly from the phrase table, and the reordering probability for each phrase pair were then integrated into our log–linear framework as an additional score.

3 N-best list rescoring

In order to easily integrate more complex models, we used n-best list rescoring in our submission. We evaluated a neural network language model using a factored representation of the words. Using this framework, we were also able to easily extend the model to a bilingual model. Furthermore, we investigated the use of an encoder-decoder model in rescoring. Finally, in cooperation with LIMSI, we used the continuous space translation models in rescoring. We used the ListNet approach as described in Section 3.4 to estimate the weights of different models in our systems.

3.1 Factored Neural Network Models

Recently, the use of neural network models in rescoring of phrase-based machine translation has shown to lead to significant improvements (Le et al., 2012; Ha et al., 2015). In addition, phrase-based machine translation can profit from factored word representations (Hoang, 2007). Using POS-tags or automatic word classes often helps to model long-range dependencies (Rottmann and Vogel, 2007; Niehues and Kolss, 2009).

In this evaluation, we evaluated a combination of both. We used RNN-based language models that use a factored representation. We hoped to improve the modeling of rare words by richer word representations. In the experiments we used up to four different word factors: the word surface form, the POS tags as well as two cluster based word fac-

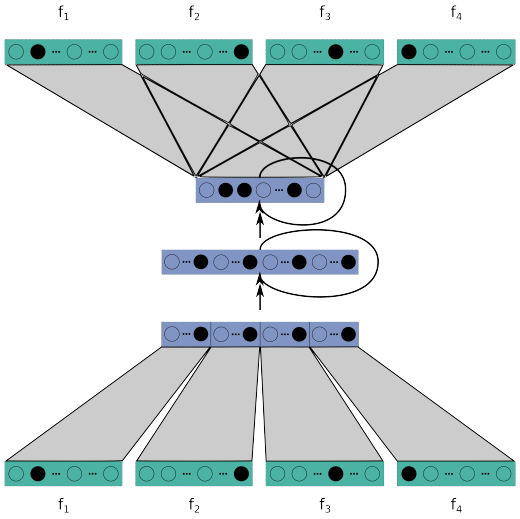

Figure 1: Factored RNN Layout

tors using 100 and 1,000 classes. The structure of the network is shown in Figure 1.

We used these word representations in the input and learnt word embeddings by using the concatenation of all word factor embeddings. On the target side, we also predicted different types of word factors.

We integrated the model into our systems by using the joint probability of all word factors as well as the individual factored probabilities as features.

Using this framework, it is straight-forward to extend it to a bilingual model which can also model translation probabilities. We achieved this by adding the word factored of the source word $s_{a(i+1)}$, that is aligned to the $i+1$ target word, to the representation of the i target word. Then we used the joint factors of the i target word and this source word to predict the $i+1$ target word. The bilingual model is referred as FactoredBM, and the language model-based is referred as FactoredLM in the evaluation section.

3.2 Recurrent Encoder-Decoder Models

The encoder-decoder architecture (Prat et al., 2001; Sutskever et al., 2014; Cho et al., 2014) has the ability of compressing all necessary information of a sequence of texts into fixed-length vectors and using this to produce an output sequence reflecting the transformation between those two sequences. Applied to machine translation, where we need to "transform" a sentence in source language to its translation in target language, the architecture has shown its usefulness. Recently, extensions of the recurrent units and the introduction

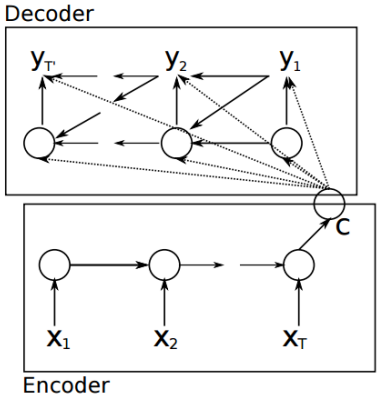

Figure 2: The recurrent encoder-decoder architecture for MT proposed by (Cho et al., 2014)

of attention mechanism allow us to train the networks to be capable of remembering longer contexts and putting decent word alignments between two sentences (Bahdanau et al., 2015; Luong et al., 2015).

Instead of using the architecture in an end-to-end fashion, which often called Neural MT (Bahdanau et al., 2015), in order to leverage other translation models that the phrase-based system produces, we opted to use it in our rescoring scheme (see 3.4).

We adapted the Neural MT framework[1] from (Luong et al., 2015) to be able to compute the conditional probability $p(f, e_i)$ in which f is the source sentence and e_i is the i^{th} translation candidate of f produced by our phrase-based decoder.

Due to the limited time, this recurrent encoder-decoder-based (ReEnDe) feature was only employed in the direction of English→German. It helped to improve considerably our translation system. We trained several ReEnDe models on the parallel EPPC and NC data, then chose the model which performed best on our development set to be used in rescoring. This model consists of 4 layers of 1000 LTSM units with the local attention and learning rate decaying mechanism similar to what the authors of the Neural MT framework were using to achieve their best single system (Luong et al., 2015).

3.3 Continuous Space Translation Models

Neural networks, working on top of conventional n-gram back-off language models (BOLMs), have been introduced in (Bengio et al., 2003; Schwenk,

[1] https://github.com/lmthang/nmt.matlab

2007) as a potential mean to improve discrete language models. More recently, these techniques have been applied to statistical machine translation in order to estimate continuous-space translation models (CTMs) (Schwenk et al., 2007; Le et al., 2012; Devlin et al., 2014). As in previous submissions, we investigated the integration of n-gram CTMs. Introduced in (Casacuberta and Vidal, 2004), and extended in (Mariño et al., 2006; Crego and Mariño, 2006), an n-gram translation model is constructed based on a specific factorization of the joint probability of parallel sentence pairs, where the source sentence has been reordered beforehand. A sentence pair is decomposed into a sequence of bilingual units called *tuples* defining a joint segmentation. The joint probability of a *synchronized* and *segmented* sentence pair can be estimated using the n-gram assumption. During training, the segmentation is obtained as a by-product of source reordering. During the inference step, the SMT decoder is assumed to output for each source sentence a set of hypotheses along with their derivations, which allow CTMs to score the generated sentence pairs.

Note that conventional n-gram translation models manipulates bilingual tuples. The data sparsity issues for this model are thus particularly severe. Effective workarounds consist in factorizing the conditional probabitily of tuples into terms involving smaller units: the resulting model thus splits bilingual phrases in two sequences of respectively source and target words, synchronised by the tuple segmentation. Such bilingual word-based n-gram models were initially described in (Le et al., 2012).

However, in such models, the size of output vocabulary is a bottleneck when normalized distributions are needed (Bengio et al., 2003; Schwenk et al., 2007). Various workarounds have been proposed, relying for instance on a structured output layer using word-classes (Mnih and Hinton, 2008; Le et al., 2011). We assume in this work the same decomposition and architecture as in (Le et al., 2012) except for the output structures.

The model is trained using the *Noise Contrastive Estimation* or *NCE* for short (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012), which only delivers *quasi-normalized*. This technique is readily applicable for CTMs. Therefore, *NCE* models deliver a positive score, by applying the exponential function to the output layer activities, instead of the more costly softmax function. Initialization is an important issue when optimizing neural networks. For CTMs, a solution consists in pre-training monolingual n-gram models. Their parameters are then used to initialize bilingual models.

Given the computational cost of computing n-gram probabilities with neural network models, a solution is to resort to a two-pass approach: the first pass uses a conventional system to produce a k-best list (the k most likely hypotheses); in the second pass, probabilities are computed by the CTMs for each hypothesis and added as new features. For this year evaluation, we used the following models: one continuous target language model and three CTMs as described in (Le et al., 2012). We also trained two versions of these four models by varying learning rate and the data resampling. We end up with 8 scores added to the k-best lists.

3.4 ListNet-based Rescoring

In order to facilitate more complex models like neural network translation models, we performed n-best list rescoring. In our experiments we generated 300-best lists for the development and test data respectively. In German→English system, we generate 3000-best list instead. We used the same data to train the rescoring that we have used for optimizing the translation system.

We trained the weights for the log-linear combination used during rescoring using the ListNet algorithm (Cao et al., 2007; Niehues et al., 2015). This technique defines a probability distribution on the permutations of the list based on the scores of the log-linear model and another one based on a reference metric. In our experiments we used the BLEU+1 score introduced by (Liang et al., 2006). Then we used the cross entropy between both distributions as the loss function for our training.

Using this loss function, we can compute the gradient and use stochastic gradient descent. We used batch updates with ten samples and tuned the learning rate on the development data.

The range of the scores of the different models may greatly differ and many of these values are negative numbers with high absolute value since they are computed as the logarithm of relatively small probabilities. Therefore, we rescaled all scores observed on the development data to the range of $[-1, 1]$ prior to rescoring.

4 Results

In this section, we present a summary of our experiments in the evaluation campaign. Individual components that lead to improvements in the translation performance are described step by step. The scores are reported in case-sensitive BLEU (Papineni et al., 2002).

In the rescoring scheme of our systems, the BLEU scores on the development set are normally smaller than those in the decoding phase because they are tuned by different optimization algorithms (ListNet and MERT). The rescoring configurations are mentioned in the tables in *italic texts*. The test scores from which we choose to be the submitted systems are mentioned in the tables in **bold numbers**.

4.1 English-German

Table 1 shows the results of our system for English→German translation task.

The baseline system consists of a phrase table extracted from all the parallel data, the word-based language models learned from all provided monolingual corpora including the large Common Crawl data. It also includes a 5-gram bilingual language model and 4-gram cluster language model trained on the monolingual part of all parallel corpora with additional information from the word alignments and 50 word classes described in Section 2.1. POS-based long-range reordering rules were applied. We used the performance in terms of BLEU on our development set to choose our combinations of features. The BLEU score of the baseline system over the test set was 22.91.

The system gained around 0.4 points on the test set in BLEU when adding lexicalized reorderings and the source-context DWLs. Both the DWLs and lexicalized reordering were trained only on EPPS and NC.

SDWL and recurrent encoder-decoder scores added into that system via the ListNet-based rescoring scheme brought considerable improvements of almost 0.9 BLEU points and this system was submitted to the conference's evaluation campaign.

On the other hands, another set of features was used in the rescoring process and helped to improve the translation performance by another 0.9 BLEU points. It included LIMSI's continuous space translation models, the factored neural network (both FactoredLM and FactoredBM) and the recurrent encoder-decoder scores. It was submitted as the joint KIT-LIMSI submission system.

System	Dev	Test
Baseline	21.81	22.91
+ DWL + Lex. Reorderings	22.44	23.34
+ *ReEnDe*	20.76	24.08
+ *SDWL + ReEnDe*	20.79	**24.21**
+ *Factored + ReEnDe + CTMs*	20.78	**24.24**

Table 1: Experiments for English→German

4.2 German-English

Table 2 shows the development steps of the German→English translation system.

The baseline system used EPPS, NC, and filtered web-crawled data for training the translation model. The phrase table was built using GIZA++ word alignment and lattice phrase extraction.

Altogether three language models were used in the baseline system, including a word-based language model, bilingual language model, and a language model built using 10M of selected data from monolingual data, based on cross entropy as described in Section 2.1. All language models were 4-gram. The word lattices are generated using short and long-range reordering rules, as well as tree-based reordering rules. A lexicalized reordering model is also included in the baseline system. We then enhanced our tree-based reordering using recursive rules. This successfully improved the translation by 0.7 BLEU points.

In this direction, we applied stemming for the German side of the corpus, inspired by (Slawik et al., 2015). Applied to the words which are *not* most frequently used $50,000$ words in the training corpus, the stemming yielded the improvement of 0.14 BLEU points.

As described in Section 2.1, we built a cluster language model using the MKCLS algorithm. Words from EPPS, NC, and the filtered crawl data were clustered into 100 different classes.

A DWL with source context increased the score on the test set slightly.

Using the additionally available monolingual data this year, we build an extra language model on words. Incorporating a big size of its training corpus, it boosts the translation performance by 0.4 BLEU points.

We then used the ListNet-based rescoring with additional models such as SDWL and Factored

LM. The rescoring is applied for $3,000$ N-best lists. The factored LM is trained for 5K of vocabulary. Finally, adding a factored BM gave another small improvement. This system was used to generate the translation submitted to the evaluation.

System	Dev	Test
Baseline	28.31	27.73
+ Resursive	28.83	28.84
+ Stem	28.83	28.98
+ MKCLS 100	28.90	29.08
+ DWL.SC	28.99	29.11
+ bigLM	28.97	29.51
+ *FactoredLM 5K* + *SDWL*	28.27	29.59
+ *FactoredBM 5K*	28.47	**29.66**

Table 2: Experiments for German→English

4.3 English-Romanian

The English→Romanian system was trained on all available parallel data and adapted to the SETimes corpus. We used pre-reordering and five language models, where two language models were word-based, two other language models were based on automatic word classes and another one was a POS-based language model. Finally, we used the DWL for this translation direction as well. The phrase-based MT system was optimized using MERT on the first half of the development set and then we generated 300-best lists.

The rescoring was optimized on the first half of the development set and on 2000 sentences from the SETimes corpus not used in training. We reported test scores on the second half of the development data.

First, we added the SDWL model in rescoring. This leads to some improvement on the development data and small improvements on the test data. Using also a factored language model and translation model could improve the translation performance by 0.7 BLEU points. We utilized a factored language model using a vocabulary of 50K words and two bilingual translation models: one with 50K word vocabulary and one with 5K words. All models used two word clusters with 100 and 1000 classes and on the Romanian side a POS factor.

5 Conclusion

In this paper, we have described the systems developed for our participation in the News Translation shared tasks of the First Conference on Statistical Machine Translation evaluation. Our systems include English→German, German→English and English→Romanian translations. All translation candidates were generated using strong baseline phrase-based systems and then rescored in combination with our new neural network-based features. We could show that the usage of neural models in rescoring significantly improved the translation.

System	Dev	Test
Baseline	39.74	29.69
+ *SDWL*	40.12	29.75
+ *FactoredRNN*	41.16	**30.57**

Table 3: Experiments for English→Romanian

Acknowledgments

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 645452.

References

Alexandre Allauzen, Lauriane Aufrant, Franck Burlot, Elena Knyazeva, Thomas Lavergne, and François Yvon. 2016. LIMSI@WMT'16 : Machine translation of news. In *Proceedings of the ACL 2016 First Conference on Machine Translation (WMT2016)*, August.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR 2015*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Zhe Cao, Tao Qin, Tie yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the 24th International Conference on Machine Learning*, pages 129–136, Corvallis, OR, USA.

Francesco Casacuberta and Enrique Vidal. 2004. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(3):205–225.

Kyunghyun Cho, Bart Van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder

for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

Josep Maria Crego and José B Mariño. 2006. Improving statistical mt by coupling reordering and decoding. *Machine Translation*, 20(3):199–215.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, Maryland.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yeh Whye Teh and Mike Titterington, editors, *Proceedings of th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 297–304.

Thanh-Le Ha, Quoc-Khanh Do, Eunah Cho, Jan Niehues, Alexandre Allauzen, François Yvon, and Alex Waibel. 2015. The kit-limsi translation system for wmt 2015. *EMNLP 2015*, page 120.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, Edinburgh, United Kingdom.

Teresa Herrmann, Jan Niehues, and Alex Waibel. 2013. Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation. In *Proceedings of the 7th Workshop on Syntax, Semantics and Structure in Statistical Translation*, Altanta, GA, USA.

Teresa Herrmann, Jan Niehues, and Alex Waibel. 2015. Source Discriminative Word Lexicon for Translation Disambiguation. In *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT15)*, Danang, Vietnam.

Hieu Hoang. 2007. Factored translation models. In *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL*, pages 868–876.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary.

Philipp Koehn, Amittai Axelrod, Alexandra B. Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of the 2nd International Workshop on Spoken Language Translation*, Pittsburgh, PA, USA.

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. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic.

Hai-Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon. 2011. Structured output layer neural network language model. In *Proceedings of ICASSP*, pages 5524–5527.

Hai-Son Le, Alexandre Allauzen, and François Yvon. 2012. Continuous space translation models with neural networks. pages 39–48, Montréal, Canada, June. Association for Computational Linguistics.

P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar. 2006. An End-to-end Discriminative Approach to Machine Translation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006)*, pages 761–768, Sydney, Australia.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrick Lambert, José A.R. Fonollosa, and Marta R. Costa-Jussà. 2006. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549.

Arne Mauser, Saša Hasan, and Hermann Ney. 2009. 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*, EMNLP '09, Singapore.

Mohammed Mediani, Eunah Cho, Jan Niehues, Teresa Herrmann, and Alex Waibel. 2011. The KIT English-French Translation systems for IWSLT 2011. In *Proceedings of the 8th International Workshop on Spoken Language Translation*, San Francisco, CA, USA.

Andriy Mnih and Geoffrey E Hinton. 2008. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, volume 21, pages 1081–1088.

Andriy Mnih and Yeh Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models.

Jan Niehues and Muntsin Kolss. 2009. A POS-Based Model for Long-Range Reorderings in SMT. In *Proceedings of the 4th Workshop on Statistical Machine Translation*, Athens, Greece.

Jan Niehues and Alex Waibel. 2013. An MT Error-Driven Discriminative Word Lexicon using Sentence Structure Features. In *Proceedings of the 8th Workshop on Statistical Machine Translation*, Sofia, Bulgaria.

Jan Niehues, Quoc Khanh Do, Alexandre Allauzen, and Alex Waibel. 2015. Listnet-based MT Rescoring. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, page 248.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1).

K. Papineni, S. Roukos, T. Ward, and W.-jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 311–318, Philadelphia, Pennsylvania.

Federico Prat, Francisco Casacuberta, and Marıa José Castro. 2001. Machine translation with grammar association: Combining neural networks and finite state models. In *Proceedings of the Second Workshop on Natural Language Processing and Neural Networks*, pages 53–60.

Anna N. Rafferty and Christopher D. Manning. 2008. Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines. In *Proceedings of the Workshop on Parsing German*.

Kay Rottmann and Stephan Vogel. 2007. 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*, Skövde, Sweden.

Helmut Schmid and Florian Laws. 2008. Estimation of Conditional Probabilities with Decision Trees and an Application to Fine-Grained POS Tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, United Kingdom.

Holger Schwenk, Marta R. Costa-jussa, and Jose A. R. Fonollosa. 2007. Smooth bilingual n-gram translation. pages 430–438, Prague, Czech Republic.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech and Language*, 21(3):492–518, July.

Isabel Slawik, Jan Niehues, and Alex Waibel. 2015. Stripping adjectives: Integration techniques for selective stemming in smt systems. *Proceedings of the 18th Annual Conference of the European Association for Machine Translation (EAMT 2015)*.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, Denver, CO, USA.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Venugopal, Andreas Zollman, and Alex Waibel. 2005. Training and Evaluation Error Minimization Rules for Statistical Machine Translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, Ann Arbor, MI, USA.

Stephan Vogel. 2003. SMT Decoder Dissected: Word Reordering. In *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*, Beijing, China.