The Karlsruhe Institute of Technology Systems for the News Translation Task in WMT 2017

Ngoc-Quan Pham, Jan Niehues, Thanh-Le Ha,
Eunah Cho, Matthias Sperber, Alexander Waibel
Karlsruhe Institute of Technology, Karlsruhe, Germany
`firstname.lastname@kit.edu`

Abstract

We present our experiments in the scope of the news translation task in WMT 2017, in three directions: German→English, English→German and English→Latvian. The core of our systems is the encoder-decoder based neural machine translation models , enhanced with various modeling features, additional source side augmentation and output rescoring. We also experiment various methods in data selection and adaptation.

1 Introduction

We participate in the WMT 17 shared task on news translation with three directions: English-German, German-English and English-Latvian. The core of our submissions is the neural attentional encoder-decoder model, which we enhanced with different features such as context gates for more efficient attention and the coverage vector for maintaining attentional information during translation. Several techniques to integrated additional information into the source text have be investigated: Pre-translation with statistical systems, mono-lingual data and phrase-table entries. Finally, we combined different models using n-best lists reranking.

2 Data

This section describes the preprocessing steps for the parallel and monolingual corpora for the language pairs involved in the systems as well as the data selection methods investigated.

2.1 German↔English

As parallel data for our German↔English systems, we used Europarl v7 (EPPS), News Commentary v12 (NC), Rapid corpus of EU press releases, Common Crawl corpus, and simulated data. Except for the common crawl corpus, no special preprocessing was applied, but only tokenization and true-casing. For the common crawl corpus, we applied noise filtering using SVM as shown in Mediani et al. (2011). Around 900K sentence pairs are filtered out using this technique.

Synthetic data is motivated by Sennrich et al. (2015a). In order to exploit the monolingual data, we used the back-translation technique. We randomly select sentences from the data as much as our parallel data, and translate them with an inverse NMT system from the target to the source language. We use this synthetic data as an additional parallel training data. Summing all corpora, the preprocessed and noise-filtered parallel data reaches 8.3M sentences for each language.

For German monolingual data, we use News Crawl data. For English, we use News Crawl and News Discussions corpus. Same as for parallel data, only tokenization and true-casing are applied.

Once the data is preprocessed, we applied byte-pair encoding (BPE) (Sennrich et al., 2015b) on the corpus. In this work, we deploy two different operation sizes, 40K and 80K.

2.1.1 Monolingual data selection

We experimented with using domain adaptation techniques to select monolingual data for back-translation. In particular, we concatenated all news-test data sets up until 2013 to form our in-domain corpus, and used news-shuffle as background data. We used the method by Axelrod et al. (2015), a class-based extension of the widely used cross-entropy difference based data selection method by Moore and Lewis (2010). For word clustering, we used Clustercat (Dehdari et al., 2016) with 20 classes. We selected an amount of data equal to the available bilingual training data. Backtranslation was done as in (Sennrich et al., 2015a). We attempted this approach for both systems with English and German as target language. However, we did not observe any improvements over selecting monolingual data at random, and did not employ this method for our final system.

2.1.2 Parallel data selection

From previous MT evaluation campaigns (Cho et al., 2016), we notice that NMT systems work well when we do fine tuning on in-domain data after training our models on out-of-domain data. Since a clear in-domain corpus is not available in this task, we conducted parallel data selection experiments to build an in-domain corpus.

We followed the approach described in (Peris et al., 2016) to extract an in-domain data set from News Commentary corpus. More specifically, an LSTM-based neural network was utilized to classify every sentence in the general corpus whether we should include it into the in-domain corpus or not. The network is trained using a "golden" corpus as the in-domain one. We took the WMT development sets from 2008 to 2013, c.a. 16K sentence pairs, to be the golden corpus for this training. The outcome is the merge of the development sets and the selected sentences from News Commentary, resulting in c.a. 100K sentence-pair in-domain corpus.

2.2 English→Latvian

The parallel corpus English-Latvian contains 2.9 million sentences which are proprocessed by TILDE[1] with language specific tokenizers. The Latvian text is only true-cased on the first letter of the sentence. We also further clean the data by using the language detection library Shuyo (2010) and remove the lines that the target sentences cannot be recognized as Latvian by the tool, resulting in about 25K sentences removed. Aside from the main data provided by the organizer, we exploit the synthetically translated monolingual data (only the news2016 part), which is provided by University of Edinburgh with a Moses phrase-based system. The training data used for the final system consists of 5 million sentences in total. For validation, we use the the first $2,000$ sentences of the Leta corpus (the rest included in the training data) and use the newsdev2017 set ($2,003$ sentences) for testing. We train a BPE (Sennrich et al., 2015b) model on the training data (including the back-translated part) with 40K operations, which is potentially helpful for a morphologically rich target language.

3 NMT Frameworks

Our systems consist of multiple neural encoder-decoder models trained using two different toolkits.

3.1 Nematus

We initially used the `nematus`[2] toolkit, in which we used the hyperparameters following previous works (Sennrich et al., 2017): minibatch size of 80, maximum sentence length of 50, word embedding size of 650, a one layer GRU with size 1,024 in the encoder and a conditional GRU decoder with hidden layer size 1,024. The gradients are scaled with norm of 1.0 and the gradient update method being used is Adam (Kingma and Ba, 2014) with learning rate 0.0001. Models are trained until the BLEU score on the validation set stops increasing. Checkpoints are saved every 20K iterations.

3.2 OpenNMT

We also employed the Torch-based (Collobert et al., 2011) toolkit OpenNMT (Klein et al., 2017) [3]. All models trained with this toolkit have two LSTM layers of 1,024 units each, and we also use the input-feeding method as described in (Luong et al., 2015). For optimization, the gradients are scaled at 5, and we experimentally use Adam with a high learning rate of 0.001 and then reduce it to 0.0005 when the perplexity of the model does not decrease anymore. Checkpoints are saved every epoch (all of the sentences are seen). We also enhanced the toolkits with different features, namely the Context Gate for attentional model (Tu et al., 2016a) and using coverage information during learning to translate (Tu et al., 2016b; Sankaran et al., 2016).

3.2.1 Context gates for machine translation

In conditional language models such as neural machine translation, the decoder makes prediction based on two sources of input: the decoder input at the current time step and the context vector queried by the attentional model. As analysed by (Tu et al., 2016a), it could be beneficial for the

[1] www.tilde.com

[2] `https://github.com/rsennrich/nematus`

[3] Our implementation for the WMT project can be found at `https://github.com/isl-mt/OpenNMT`

translation model to be able to control the influence of each prediction source. Concretely, inadequate translation can happen due to the bias over the current decoder input. We followed the authors to integrate a soft gating mechanism to alleviate this problem. Specifically, in our neural translation model, given the target hidden state h_t and the source context vector c_t, an attentional hidden state is formed by concatenation (Luong et al., 2015).

Alternatively, we use h_t and c_t to learn a soft context mask that prevents the activation of both states. The mentioned states are then masked with learned gates, and concatenated before being fed into the final linear regression layer.

Note that the authors (Tu et al., 2016a) built their model on top of the conditional GRU based network from Bahdanau et al. (2014), while ours are essentially an multi-layer LSTM decoder with an additional attention layer. Such difference leads to the minor change in terms of implementation, which may not replicate the same improvement as the original work.

3.2.2 Coverage mechanism for attention model

Various works have pointed out that the attention neural machine translation model can be benefit by constraining the attentional process to adequately cover the source words (Sankaran et al., 2016; Tu et al., 2016b; Mi et al., 2016; Luong et al., 2015). Different proposals share similar ideas which is to incorporate alignment information from the previous time steps into the attentional neural network. Our experiment inherits the neural fertility model from (Tu et al., 2016b) which uses an explicit vector to keep track of the alignment information. At every time step, the network makes an attentional decision with the help of the coverage vector, which is in turn updated using the alignment vector and the source context with a simple Gated Recurrent Unit (GRU).

4 Integration of Additional Resources

In this section, we show several techniques we applied in order to integrate additional resources into the translation. First, we integrate monolingual information using a multi-lingual NMT approach. In addition, we extracted information from PBMT systems.

4.1 Monolingual Data

When the encoder of an NMT system of a well-chosen architecture considers words across different languages, the model is expected to learn a good representation of the source words in a joint embedding space, in which words carrying similar meaning would have a close distance from each other. In turn, the shared information across source languages could help improve the choice of words in the target side. For example, the word *Flussufer* in German and the word *bank* in English should be projected in the joint embedding space in close proximity. This information might help to choose the French word *rive* over *banque*.

To make an attention NMT for single language pair translation to support a multilingual NMT that shared the common semantic space, (Ha et al., 2016b) suggested language-specific coding. Basically, language codes are appended to every word in source and target sentences and indicate the original language of the word. This information will be then passed to the training process of the NMT system. For example, an English-German sentence pair *excuse me* and *entschuldigen Sie* become *_en_excuse _en_me* and *_de_entschuldigen _de_Sie*. By doing so, they can train a single multilingual system that translates from several source languages to one or several target languages. When we have n English-German sentence pairs and m French-German sentence pairs, for example, we can train a single NMT system with a parallel corpus of $n + m$ sentence pairs. Then we can use the trained model to either translate from English or from French to German.

The aforementioned multilingual NMT can be used wisely as a novel way to utilize the monolingual data, which is not a trivial task in NMT systems. Particularly, if we want to translate from English to German, we can use a corpus in German as an additional German-German data similar to the way we utilize the French-German parallel corpus. Thus, the encoder is shared between the source and the target languages (English and German), and the attention is also shared across languages to help the decoder selects better German words in the target side. The system implemented this idea is referred as a *mix-source* system.

For this evaluation, we apply the idea of that multilingual NMT approach in the English-German direction in order to make use of the German monolingual corpus and gain additional improvements.

4.2 Pre-translation

One of the main problems of current NMT system is its limited vocabulary (Luong et al., 2014), generating difficulties when translating rare words. While the overall performance of NMT is significantly better on many tasks compared to SMT (Bojar et al., 2016), the translation of words seen only a few times is often not correct. In contrast, PBMT is able to memorize a translation it has observed only once in the training data. Therefore, we tried to combine the advantages of NMT and PBMT using pre-translation as described in (Niehues et al., 2016).

In the first step, we translate the source sentence f using the PBMT system generating a translation e^{SMT}. Then we use the NMT system to find the most probable translation e^{*} given the source sentence f and the PBMT translation e^{SMT}. Thus, we create a mixed input for the NMT system consisting of both sentences by concatenating them. This scheme, however, may lead to errors when the source and target languages have a same word in surface, but with different meanings, e.g. *die* in English is a verb, while it is an article in German. In order to prevent such errors, we use a separate vocabulary for each language. Using the BPE of the input (Sennrich et al., 2015b), we are able to encode any input words as well as any translation of the PBMT system. Thereby, the NMT is able to learn to copy translations of the PBMT system to the target side. The pre-translation method is applied on the German → English direction.

4.3 Integration of Selected Phrase Pairs

One main drawback of the aforementioned approach is that all training data as well as the test data has to be translated using a phrase-based MT system. Therefore, this is a time-consuming approach.

In a second approach to integrate information for rare words from the phrase-based MT system, we relied only on the phrase table. Using this technique, we annotate rare words with their possible translation according to the phrase table. In the first step, we need to identify the words for which we want to provide a possible translation. Then we need to select a translation from the phrase table and, finally, we need a method to provide the translation of the word optional to the NMT system.

In our approach, we consider all words that were split into several words by the byte pair encoding as rare words. For these words, we search their possible translations in the phrase table. We took the phrase pair with the longest source phrase that covers the word. If there are several translation options for this source phrase, we select the one where the log-sum of all fours probabilities in the phrase table is the highest.

We integrate this information into the source sentence, by appending the source phrase and the translation from the phrase table. We also annotate the beginning and end of the phrase with a special character. When we have the source sentence *Obama empfän@@ gt Netanyahu* and a phrase pair *empfän@@ gt* |||||| *receives* in the phrase table, we will generate the following input for the NMT system: *Obama # empfän@@ gt ## receives # Netanyahu*

5 System Combination

Combination of different neural networks often leads to better performance, as shown in various applications of neural networks and previous NMT submissions in evaluation campaigns (Bojar et al., 2016). A successful technique is to ensemble different checkpoints of a model or models with different random initialization. While this is a very helpful technique, it has a potential drawback that it can only be performed easily for models using the same input and output representations.

In order to further extend the variety of models, we combine the output of several ensemble models by an n-best list combination. A first approach is to generate an n-best list from all or several of the models. Afterwards, we combine the n-best lists into a single one by creating the union of the n-best lists. Since every model only generated a subset of the joint list, we rescored the joint list by each model. Finally, we used a combination of all the scores to select the best entry for every source sentence. In previous work (Cho et al., 2017), it was shown that it is often sufficient to use the n-best list of the best model and rescore this n-best list with the different models. In our experiments, we used $n = 50$ for the n-best list size.

For systems to be combined, we use the NMT system generated by different frameworks (described in Section 3), as well as the pre-translation and multi-lingual systems (described in Section 4). We also combine systems using different BPE sizes. In addition, we use a system that generates the target sentence in the reversed order (Sennrich et al., 2015a; Liu et al., 2016; Huck et al., 2016).

After joining the n-best lists and rescoring it using the different systems, we have k scores for every entry in the n-best lists. In our experiments, we use two different techniques to combine the scores. The first method is to use the sum of all scores. Especially, if the performance of the different models is similar, we do not need to weigh the different models. Similar to the ensemble methods we can reach a good performance by using equal weights. In a second approach, we use the ListNet algorithms (Cao et al., 2007; Niehues et al., 2015) to find the optimial weights for the individual models.

5.1 ListNet-based Rescoring

In order to find the optimal weights for the different models, we use 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 individual models and another one based on a reference metric. In this set of experiments, we use the BLEU+1 score introduced by (Liang et al., 2006). Then we measure the cross entropy between both distributions as the loss function for our training. We trained the weights for the different models on the validation set also used during training the NMT systems.

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

The range of the scores of the different toolkits may greatly differ. Therefore, we rescaled all scores observed on the development data to the range of $[-1, 1]$ prior to rescoring.

6 Results

In this section, we describe the systems used to generate the final hypothesis for official test set. We participated in German→English, English →German, and English→Latvian translation tasks.

6.1 German→English

All German to English translation system are trained on the parallel data as well as back-translated data (Sennrich et al., 2015a) randomly selected from the monolingual news data. We use newstest2013 as validation data. Using this data, we train our initial system with the Nematus toolkit and a byte pair encoding size of 40K operations (*Nematus 40K*). The translation for all *Nematus* based systems are generated with ensembled system of different checkpoints. Although we also attempted to select the data for backtranslation as described in Section 2.1, initial experiments did not show improvements on the translation quality. Therefore, we use the randomly selected data for the remaining experiments.

In addition, we build a system with a reverse target order (R2L) (Liu et al., 2016) and the pre-translation. The pre-translation was generated by the PBMT system used in WMT 2016 (Ha et al., 2016a). Both performed slightly better than the baseline system.

When increasing the size of BPE operation to 80K, we observe the improvements on the translation quality, by 1.4 BLEU points.

In addition to Nematus, we also used the OpenNMT framework to build a network. For this language pair, we used the context gate, but not the coverage model. In contrast to the Nematus based systems, we did not ensemble different checkpoints. When using OpenNMT this technique did not yield an improvement in translation performance. When OpenNMT is trained using 40K BPE units (single system), we reach a BLEU score of 38.39. The default architecture of OpenNMT - utilizing two hidden layers - is deemed to be one reason for its outstanding performance.

In addition, we build a system using rare words annotated with their translations. In contrast to the baseline OpenNMT system, this configuration utilizes only half the hidden size. For comparison, a baseline system using this hidden size achieved a BLEU score of 36.91 on newstest2016. Although we did not improve the performance over the baseline, it was beneficial to use the system in the combination.

Finally, we generated an n-best list using the best performing system OpenNMT 40K. Then we used all the other models to rescore this n-best lists. The scores are combined linearly. The weights were optimized using the ListNet algorithm on newtest 2015. This resulted to the best performance of 39.10. The combination of all models improve the translation performance by another 0.7 BLEU points.

System	News2015	News2016
Nematus 40K	29.64	35.96
R2L		36.67
PreMT		36.86
Nematus 80K		37.38
OpenNMT 40K	31.48	38.39
RareWords	29.73	36.50
ListNet	32.33	**39.10**

Table 1: Experiments for German→English

System	Baseline	Adapted
Forward Normal	30.20	31.27
Forward Context Gate	30.44	31.36
Forward Ensembled	30.68	32.22
R2L	30.54	31.56
Mix-source	30.11	31.11
Pre-translation	30.67	-
Rescoring	-	**33.17**

Table 2: Experiments for English→German

6.2 English→German

Table 2 shows the results of the English→German translation task. The scores are reported in BLEU scores and evaluated on test2016. We used OpenNMT framework on the preprocessed data (parallel, sampled, back-translated as in Section 2.1). For all experiments, we used BPE operation at 40K.

The systems differ in the training method and the architectures. In the first series of experiments *Forward*, training sentences are seen in their natural direction (left-to-right in this case). For this type of experiments, we trained with two architectures: normal and with context gates. The *Context Gate* system got a small improvement over the normal one. The two architectures share the same vocabularies and ensembling them helped us to get more improvements. In the second series of experiments *R2L* the target sentences were reversed in order (right-to-left). And the third type is the mix-source systems described in Section 4.1 and in (Ha et al., 2016b). In addition, we also used a pre-translation system. The systems have different vocabularies and they were eventually combined using our ListNet-based rescoring (Section 5.1).

For each type of experiments, we conducted fine tuning on the small in-domain corpus mentioned in Section 2.1.2, and the best adapted model based on its BLEU score on test2015 was picked for the ensembling andor rescoring. In all systems except for pre-translation, we observed considerable improvements, around 1 BLEU point, when applying fine tuning (c.f. *Adapted* column).

Finally, we rescored and combined four adapted systems (*Forward Ensembled*, *R2L*, *Mix-source* and *Pre-translation*) to get our submission system to the campaign. It achieved 33.17 BLEU points on test2016, 0.9 BLEU points better than the *Forward Ensembled* system and 1.6 BLEU points better than our best single system (*R2L*).

6.3 English→Latvian

The result of the English→Latvian translation task is illustrated in table 3. Our baseline models are trained with both frameworks using the concatenation of the actual parallel and back-translated data. We use dropout of 0.2 for both frameworks. For Nematus, the convergence was seen after about $540K$ iterations (about 9 epochs), with the best validation and test BLEU score achieved of 19.92 and 22.95 respectively. With OpenNMT, we obtained 20.62 and 24.11 BLEU points for the validation and test set, after training for 8 epochs (4 with high learning rate of 0.001, 2 with 0.0005 and last 2 with 0.00025.

Regarding the two enhancement features mentioned above, the simple Context Gate improved the scores by 0.2 and 0.6 on the two sets respectively, while integrating the coverage mechanism in the attention model yields a further 1.1 and 0.5 BLEU scores. The decoder recurrent network has always received previous context information through input-feeding. Surprisingly, the coverage vector still manages to improve the model performance. We assume that the gain comes from a stronger attention network, which has more parameters than the cosine similarity between the hidden state and the context, and the fact that the coverage vector can maintain a longer past attentional information compared to input-feeding.

It is notable that even though the improvement has been observed, it is not consistent throughout the sets. One possible explanation is the difference between the development (from Leta) and the test set (from news) in terms of domain and difficulty.

Regarding the consistency between BLEU score and perplexity, the model with higher BLEU score does not necessarily have lower perplexity (across different settings, for example baseline vs. coverage) even though we choose the model with the best perplexity for reporting BLEU scores.

This is the case even when these models share the same vocabulary. We can see that perplexity is a good measure to choose models within a single run, even though it is not informative to compare models with different network topologies.

By ensembling the three models, we managed improving the translation performance by 1.3 BLEU points. Our final submission is done by using another model trained with reversed target sentences to rescore the n-best list ($n = 20$) generated by the ensembled system, which improves about 0.4 BLEU.

System	LetaDev	News2016
Nematus 40K	19.92	22.95
OpenNMT 40K	20.62	24.11
+ Context Gate	20.88	24.71
+ Coverage Mode	21.91	**25.20**
Ensemble (3 models)	-	**26.54**
+ Reranking R2L	-	**26.96**

Table 3: Experiments for English→Latvian

7 Conclusion

In conclusion, we described our experiments in the news translation task in WMT 2016, in which we attempted to try out several techniques across different language pairs. The model-wise modifications such as context gate and coverage provided slight improvement, while we find out that NMT models can benefit greatly from adaptation and pre-translation. As observed in previous works, the most consistent gain mostly comes from system ensembling/combination and reranking.

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. The research by Thanh-Le Ha was supported by Ministry of Science, Research and the Arts Baden-Württemberg. The work by Jan Niehues was supported by the Carl-Zeiss-Stiftung.

References

Amittai Axelrod, Yogarshi Vyas, Marianna Martindale, and Marine Carpuat. 2015. Class-Based N-gram Language Difference Models for Data Selection. In *IWSLT (International Workshop on Spoken Language Translation)*. pages 180–187.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation (wmt16). In *Proceedings of the First Conference on Machine Translation (WMT)*. volume 2, pages 131–198.

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*. Corvallis, OR, USA, pages 129–136.

Eunah Cho, Jan Niehues, Thanh-Le Ha, Matthias Sperber, Mohammed Mediani, and Alex Waibel. 2016. Adaptation and combination of nmt systems: The kit translation systems for iwslt 2016. In *Proceedings of the 13th International Workshop on Spoken Language Translation (IWSLT 2016)*.

Eunah Cho, Jan Niehues, Thanh-Le Ha, and Alexandre Waibel. 2017. Analyzing neural mt search and model performance. In *Proceedings of The First Workshop on Neural Machine Translation. Association ofr Computational Linguistics*.

Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. 2011. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*. EPFL-CONF-192376.

Jon Dehdari, Liling Tan, and Josef van Genabith. 2016. BIRA: Improved predictive exchange word clustering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*. Association for Computational Linguistics, San Diego, CA, USA, pages 1169–1174. http://www.aclweb.org/anthology/N16-1139.pdf.

Thanh-Le Ha, Eunah Cho, Jan Niehues, Mohammed Mediani, Matthias Sperber, Alexandre Allauzen, and Alexandre Waibel. 2016a. The karlsruhe institute of technology systems for the news translation task in wmt 2016. In *Proceedings of the First Conference on Machine Translation, Berlin, Germany. Association for Computational Linguistics*.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016b. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798* .

Matthias Huck, Alexander Fraser, and Barry Haddow. 2016. The edinburgh/lmu hierarchical machine translation system for wmt 2016. In *Proc. of* *the ACL 2016 First Conf. on Machine Translation (WMT16), Berlin, Germany, August*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints* .

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)*. Sydney, Australia, pages 761–768.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of NAACL-HLT*. pages 411–416.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* .

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation.

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.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. *arXiv preprint arXiv:1605.03148* .

Robert C Moore and William Lewis. 2010. Intelligent Selection of Language Model Training Data. In *Proceedings of ACL*.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *the 26th International Conference on Computational Linguistics (Coling 2016)*.

Jan Niehues, Quoc Khanh Do, Alexandre Allauzen, and Alex Waibel. 2015. Listnet-based MT Rescoring. *EMNLP 2015* page 248.

Álvaro Peris, Mara Chinea-Rios, and Francisco Casacuberta. 2016. Neural networks classifier for data selection in statistical machine translation. *arXiv preprint arXiv:1612.05555* .

Baskaran Sankaran, Haitao Mi, Yaser Al-Onaizan, and Abe Ittycheriah. 2016. Temporal attention model for neural machine translation. *arXiv preprint arXiv:1608.02927* .

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Valencia, Spain, pages 65–68. http://aclweb.org/anthology/E17-3017.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany.

Nakatani Shuyo. 2010. Language detection library for java.

Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. 2016a. Context gates for neural machine translation. *arXiv preprint arXiv:1608.06043* .

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016b. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811* .