The RWTH Aachen University Machine Translation Systems for WMT 2019

Jan Rosendahl, Christian Herold, Yunsu Kim, Miguel Graça, Weiyue Wang, Parnia Bahar, Yingbo Gao and Hermann Ney
Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
`<surname>@i6.informatik.rwth-aachen.de`

Abstract

This paper describes the neural machine translation systems developed at the RWTH Aachen University for the De→En, Zh→En and Kk→En news translation tasks of the *Fourth Conference on Machine Translation* (WMT19). For all tasks, the final submitted system is based on the Transformer architecture. We focus on improving data filtering and fine-tuning as well as systematically evaluating interesting approaches like unigram language model segmentation and transfer learning. For the De→En task, none of the tested methods gave a significant improvement over last years winning system and we end up with the same performance, resulting in 39.6% BLEU on `newstest2019`. In the Zh→En task, we show 1.3% BLEU improvement over our last year's submission, which we mostly attribute to the splitting of long sentences during translation. We further report results on the Kk→En task where we gain improvements of 11.1% BLEU over our baseline system. On the same task we present a recent transfer learning approach, which uses half of the free parameters of our submission system and performs on par with it.

1 Introduction

The RWTH Aachen University developed three systems for the German→English, Chinese→English and Kazakh→English WMT19 news translation tasks.

For the language pairs De→En and Zh→En there is a lot of training data available, however it consists partially of low quality data. Therefore we improve our data filtering techniques and the preprocessing of the data. We also studied different settings for the fine-tuning and ensembling steps of the final models.

For the low resource Kk→En task we furthermore make use of additional Ru−En/Kk parallel data, exploiting the similarities between the Russian and Kazakh languages.

This paper is organized as follows: In Section 2, we describe our data preprocessing. Our translation software and baseline setups are explained in Section 3. The results of the experiments for the various language pairs are summarized in Section 4.

2 Preprocessing

For English, German and Kazakh data, we use a simple preprocessing pipeline consisting of minor text normalization steps (such as removing some special UTF-8 characters), followed by frequent casing from the Jane toolkit (Vilar et al., 2010). We remove all the spaces in the Chinese data and applied a dictionary to convert traditional to simplified Chinese characters (including quotation marks). The Kk→En experiments also use the Moses tokenizer (Koehn et al., 2007) as an intermediate step.

In this work, we consider two variants of byte-pair encoding (BPE): (i) the original approach as proposed by Sennrich et al. (2016) (further denoted as pure BPE) and (ii) the unigram language model (ULM) approach by Kudo (2018) (further denoted as ULM-BPE). We apply the ULM implementation from Kudo and Richardson (2018) (SentencePiece) to segment words into subwords for De→En and Zh→En (Kudo, 2018). The segmentation model is trained jointly for the De→En task with a vocabulary size of 50k, and it is trained separately for the Zh→En task with a vocabulary size of 32k. For De→En, we use data from CommonCrawl, Europarl, NewsCommentary and Rapid. For Zh→En, we use 12M out of the 25M sentence pairs to train the segmentation model. When applying the ULM-BPE model, we employ a 30-best list for Chinese→English and try differ-

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers (Day 1) pages 349–355
Florence, Italy, August 1-2, 2019.

ent n-best sizes for German→English explained in Section 4. For Kk→En, we use joint pure BPE with 50k operations unless otherwise stated.

3 MT Systems

The final systems submitted by RWTH Aachen are based on the Transformer architecture implemented in the Sockeye sequence-to-sequence framework for neural machine translation (NMT) (Hieber et al., 2017) which is built on top of MXNet (Chen et al., 2015).

Our models resemble the 'big' architecture as presented by Vaswani et al. (2017) consisting of 6 layers in both encoder and decoder with 16 heads in all multi-head attention layers. We train our models using the Adam optimizer (Kingma and Ba, 2014) with a learning rate ranging from 0.0001 and 0.0003. We employ a learning rate scheduling scheme which scales down the learning rate if no improvement in perplexity on the development set has been observed for several consecutive evaluation checkpoints. A warmup period with constant or increasing learning rate was not used. During training we apply dropout ranging from 0.1 to 0.3. All batch sizes are specified on the token level and are chosen to be as big as the memory of the GPUs allows. In case of the utilization of multiple GPUs we use synchronized training, i.e. we increase the effective batch size. In the Kk→En scenarios, the parameters of the word embeddings and output layer projection are shared and 8 attention heads are used throughout the model.

Our fine-tuning strategy involves re-starting training with a lower learning rate on an in-domain data set, using the optimal parameters from the larger data set as initialization.

We perform experiments using the workflow manager Sisyphus (Peter et al., 2018).

4 Experimental Evaluation

In this section, we present our results on the three translation tasks in which we participated. We report case-sensitive BLEU (Papineni et al., 2002) scores as well as results on the TER (Snover et al., 2006) and CTER (Wang et al., 2016) measures. All reported scores are given in percentage and the specific options of the tools are set to be consistent with the calculations of the organizers.

			newstest2015 (dev)		
Segmentation	**n_best**	**vocab**	BLEU	TER	CTER
pure BPE	-	≈ 50k	32.1	54.2	50.2
ULM-BPE	10	20k	32.2	54.1	49.5
	10	30k	32.2	54.2	49.5
	10	50k	32.2	54.3	49.7
	30	50k	**32.6**	**52.8**	**49.2**
	120	50k	32.2	54.2	49.4
+ not joint	10	50k	31.9	54.7	49.9

Table 1: Results in percentage of our comparison of the ULM-BPE to pure BPE on the De→En task. If not stated otherwise the operations are learned jointly.

4.1 German→English

For experiments on the De→En task we use the Transformer architecture as described in Section 3 with `newstest2015` as the development set. We compare the performance of the SentencePiece implementation of the ULM-BPE to that of pure BPE. For these experiments, we train a system using the same architecture as the 'base' Transformer (see Vaswani et al. (2017)), but without tied embedding weights, on the data from CommonCrawl, Europarl, NewsCommentary and Rapid i.e. about 6M sentence pairs. We train a baseline with 50k pure joint BPE merge operations same as last year's winning system and try different vocabulary and nbest sizes for the segmentation based on a unigram language model. As can be seen in Table 1, there are only minor differences in performance. For all follow-up experiments, we use a segmentation based on the unigram language model from the SentencePiece segmenter with a vocabulary size of 50k and unigram language model with a 30-best list since it performs best with an improvement of 0.5% BLEU over the pure BPE baseline.

The main results of the De→En task are presented in Table 2. We start with a 'base'Transformer on all parallel data except the ParaCrawl resulting in a BLEU score of 32.6% on `newstest2015`.

We filter ParaCrawl based on the word-to-token ratio, average-word-length, source-target-length ratio, and source-target Levenshtein distance measures as presented in Rossenbach et al. (2018). The remaining corpus of 23M sentence pairs is scored using a count-based KenLM (Heafield, 2011) 5-gram language model on the target side and we select the top 50% as described by Schamper et al. (2018).

We train a 'big' Transformer in the En→De direction and back-translate the deduplicated NewsCrawl 2018 monolingual corpus. This back-translation system is trained on CommonCrawl, Europarl, NewsCommentary, Rapid and on the 23M sentence pairs from the filtered version of ParaCrawl as well as on 18M synthetic sentence pairs from a back-translated NewsCrawl 2017 corpus. It achieves 31.3% BLEU and 29.9% BLEU on the En→De task on `newstest2015` and `newstest2017` respectively.

To filter out sentence pairs that were copied instead of translated by the system, we apply a filtering method based on the Levenshtein distance between source and target sentences (Rossenbach et al., 2018). This has further reduced the synthetic corpus size to 15.9M sentence pairs which are used to train our final systems.

We oversample CommonCrawl, Europarl, NewsCommentary and Rapid by a factor of 3 and end up with a corpus of roughly 47M lines (18M oversampled, 1M Wikititles, 16M synthetic, 11M ParaCrawl). Training a 'big' Transformer on this corpus leads to a performance of 36.3% BLEU on the dev set as is shown in Table 2. Finetuning on the test sets from previous years (excluding only `newstest2015` and `newstest2017`) adds another 0.9% BLEU. We train two models with this configuration and experiment with different ensembles. For our final submission we pick the 3 best checkpoints out of the 2 training runs, apply finetuning to them and use a linear ensemble of them for decoding with a beam size of 12. The final performance of the ensemble is 37.4% BLEU on the dev set and 39.6% BLEU on `newstest2019`.

4.2 Chinese→English

The original Chinese-English training set contains 25.8M sentence pairs. After applying the preprocessing steps described in Section 2, we first filter out 1.1M sentence pairs which contain a large number of illegal characters (on either side). This step is performed using a Gaussian mixture model, which uses UTF-8 blocks as feature vectors and is trained on the Chinese and English development data sets. Then we apply deduplication on both sides, which further removes around 5.8M sentence pairs. From the remaining 18.9M sentence pairs we sampled 12M sentences from each side to follow the SentencePiece approach as described in Section 2. Note that we did not use any additional tools to pre-segement the Chinese data.

We also use the provided Chinese and English monolingual data and apply the same preproceesing procedure. After the filtering, the Chinese and English monolingual data sets contain 27.5M and 52.9M sentences respectively. We train LSTM-based Chinese and English language models on these monolingual data sets, as well as a big Transformer-based Chinese→English translation model on the 18.9M bilingual data set. Note that here the Chinese language model uses characters and the English language model uses sub words. The concatenation of the `newsdev2017` and `newstest2017` data sets are used as the development set for training. Then we apply the language models to score the Chinese and English training sentence pairs. The translation model is used to decode the entire training set and then we calculate the CHRF score (Popović, 2015) of each hypothesis. Then the remaining 18.9M sentence pairs are further filtered according to the language model perplexities and CHRF scores. Only sentence pairs that satisfy the following three conditions are retained:

- The CHRF score is higher than 0.55;
- The Chinese language model log-perplexity is lower than 5;
- The English language model log-perplexity is lower than 7.

Only about 13.7M parallel sentence pairs from the training data is retained after this round of filtering.

The English language model is also used to score the English monoligual data. We randomly sub-sample 10M English sentences from the filtered monolingual data for back-translation. The synthetic data is generated by a big Transformer-based En→Zh translation model trained on the 18.9M sentence pairs, i.e. before the last round of filtering.

We train the following Transformer-based translation models on the final 23.7M parallel sentences (each batch contains 4k tokens if not stated):

1. Transformer big architecture (Vaswani et al., 2017);
2. Transformer big architecture with 7 encoder and 7 decoder layers, gradient accumulation of 2 batches, which yields an effective batch size of 8k tokens;

		newstest2015 (dev)			newstest2017			newstest2019		
	Systems	BLEU	TER	CTER	BLEU	TER	CTER	BLEU	TER	CTER
1	Transformer Base	32.6	53.7	49.2	33.8	53.0	49.9	35.7	52.2	49.9
2	Transformer Big + Paracrawl + BT	36.3	50.2	45.5	38.3	48.9	45.8	37.5	50.7	46.6
3	+ fine-tuning	37.2	49.4	45.0	39.5	47.8	44.8	38.9	49.2	45.3
4	Ensemble†	37.4	49.1	44.7	39.9	47.4	44.6	39.6	48.4	44.7

Table 2: Main results for the German→English task measured in BLEU [%], TER [%] and CTER [%] †: Submitted system.

3. Transformer big architecture with gradient accumulation of 4 batches, which yields an effective batch size of 16k tokens;

4. Transformer big architecture with BLEU as metric for the learning rate reduction scheme;

5. Self-attentive encoder + LSTM decoder network (Chen et al., 2018).

All models are trained for around 14 epochs and during decoding we use a beam size of 16. As can be seen in Table 3 the first four systems show about equal performance while the LSTM decoder stays 0.4% BLEU behind the baseline on the dev set and 0.7% BLEU on `newstest2018`. Ensembling of the four strongest models provides 1.3% BLEU improvement over the baseline on the dev set.

In addition, we found that there are many long source samples in the test set. As during training we eliminate all samples which are longer than 100 subwords, our system does not perform well in the translation of longer samples. To tackle this problem, we first split all samples, which include '`.`', '`!`', '`?`' or '`;`' characters, into shorter sentences. If there are still sentences which contain more than 80 subwords, we split them on '`,`' once, in a way that keeps the lengths of the two separated sentences as equal as possible. This splitting brings up to 1.1% BLEU improvements on `newstest2018`. The final submitted system achieves a BLEU score of 31.7% on `newstest2019`.

4.3 Kazakh→English

We tackle the low-resource Kazakh→English task by leveraging additional mono- and bilingual data via back-translation, language modeling and transfer learning. Our main results are summarized in Table 4 and we deviate from the system described in Section 3 by using model dimensions of 512 and internal projections 2,048, which we further denote as the base model. A larger variant is used for Systems 4-7 with a model dimension of 1,024. A batch size of 10k words or 8k subwords is used for the smaller and larger models, respectively. This is achieved by accumulating gradients over 4 smaller batches.

In total, we leverage 24M synthetic sentence pairs and over-sample all available Kk−En data to obtain a ratio of 1:4 (authentic:synthetic) for systems 2-3 and 1:2 for systems 4-7. The Kk→En data consists of 224k training samples. For the synthetic data, we make use of the Ru-En bilingual data: the Yandex and News Commentary corpora plus 10M sentences from the UN corpus. Further, the organizers supply a crawled Kk-Ru corpus, from which we remove redundant sentences by using the technique described by Rossenbach et al. (2018). Finally, 10M sentences are sub-sampled from News Crawl 2017 for back-translation. As in-domain data, we make use of the 2014-2018 Ru-En test sets of past competitions.

The Russian side of the Kk-Ru corpus is translated to English using the small model variant and 50k joint pure BPE operations. The Russian side of the Ru-En corpus is translated to Kazakh by the former setup on the crawled corpus. Back-translations are generated using a bilingual base model (System 1), i.e. that shares parameters between both translation directions, trained with 20k joint pure BPE operations. The model itself includes 4M back-translated sentences from News Crawl 2017 and is fine-tuned on the News Commentary corpus of the Kk-Ru corpus.

We also experiment with transfer learning as presented by Kim et al. (2019). In this framework, we train a Ru→En model with non-joint pure BPE vocabularies[1] on the corresponding WMT 2018 translation task. Kazakh word embeddings are then trained on all available monolingual data,

[1] 20k operations for Russian, 50k operations for English

		dev			newstest2018	
	Systems	BLEU	TER	CTER	BLEU	CTER
1	Transformer 'big'	25.2	65.8	60.6	25.8	63.3
2	+ 7th layer + grad-acc 2	25.4	65.6	60.2	25.8	62.7
3	+ grad-acc 4	25.5	65.0	60.0	25.9	62.6
4	+ optimize on BLEU	25.4	65.6	60.6	26.0	63.3
5	+ LSTM decoder	24.8	66.3	61.4	25.5	63.5
6	Ensemble [1,2,3,4]	26.5	64.2	58.9	26.9	61.4
7	+ Split long sentences†	-	-	-	**28.0**	**60.4**

Table 3: Results for Zh→En measured in BLEU [%], TER [%] and CTER [%]. The development set is the concatenation of newsdev2017 and newstest2017. TER computation fails on newstest2018.
†: Submitted systems.

			newsdev2019			newstest2019		
	Systems	**Size**	BLEU	TER	CTER	BLEU	TER	CTER
1	Baseline	base	15.9	75.8	74.8	12.8	78.6	76.7
2	Transfer	base	21.6	72.8	64.1	23.6	69.0	62.5
3	+ fine-tuning		22.0	72.2	63.9	23.9	67.9	60.5
4	Scratch	large	21.5	72.9	64.8	23.2	68.9	62.7
5	+ fine-tuning		22.2	72.0	63.9	23.3	68.8	61.2
6	+ search tuning†		22.8	71.1	64.9	24.2	66.8	61.2
7	+ LM†		23.6	71.2	67.2	23.1	69.9	66.2

Table 4: Results measured in BLEU [%], TER [%] and CTER [%] for Kk→En. †: Submitted systems.

processed with 20k pure BPE operations, and are mapped to the same distribution as the Russian embeddings via an unsupervised mapping (Conneau et al., 2017). Finally, training is initialized with the replaced parameters and fine-tuned on the Kk-En task (System 2+3). We expect a bigger model to perform better on the Ru→En task and therefore transfer better to this task, but time constraints prohibited this.

Fine-tuning on the translated news test sets from the Ru→En task (System 5) improves performance by 0.7% BLEU on the development set but does not generalize to test set improvements. The length penalty and beam size hyperparameters were tuned to maximize the difference of BLEU and TER on newsdev2019 (System 6). Finally, we experiment with adding a 5-gram modified Kneser-Ney language model (Chen and Goodman, 1999) during inference using KenLM (Heafield, 2011) (System 7). We perform a log-linear combination and re-run the optimization grid search as before with the additional language model scaling factor. This improves the development set performance but considerably decreases the test set performance. In hindsight, our experimental setup was flawed due to not having unseen test data and therefore overfitting on the development set, clearly seen by comparing Systems 6 and 7.

5 Conclusion

This paper describes the RWTH Aachen University's submission to the WMT 2019 news translation task. For all language pairs we use the Transformer architecture. Different methods for data filtering, preprocessing and synthetic data creation were tested. We experiment with different segmentation schemes, model depth, language modelling during search and transfer learning. Our De→En system performs on par with our 2018 submission and our Zh→En model shows an 1.3% BLEU improvement over our last year's submission. For the Kk→En system we gain improvements of 11.4% BLEU over a standard semi-supervised baseline resulting in a final performance of 24.2% BLEU on newstest2019.

Acknowledgements

This work has received funding from the European Research Council (ERC) (under the European Union's Horizon 2020 research and innovation programme, grant agreement No 694537, project ”SEQCLAS”) and the Deutsche Forschungsgemeinschaft (DFG; grant agreement NE 572/8-1, project ”CoreTec”). The work reflects only the authors' views and none of the funding agencies is responsible for any use that may be made of the information it contains.

References

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 76–86, Melbourne, Australia.

Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*. Version 1.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*. Version 2.

Yunsu Kim, Yingbo Gao, and Hermann Ney. 2019. Effective cross-lingual transfer of neural machine translation models without shared vocabularies. In *Annual Meeting of the Assoc. for Computational Linguistics*, Florence, Italy.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. Version 9.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 66–75, Melbourne, Australia.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Jan-Thorsten Peter, Eugen Beck, and Hermann Ney. 2018. Sisyphus, a workflow manager designed for machine translation and automatic speech recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 84–89.

Maja Popović. 2015. CHRF: charactern-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisboa, Portugal.

Nick Rossenbach, Jan Rosendahl, Yunsu Kim, Miguel Graça, Aman Gokrani, and Hermann Ney. 2018. The rwth aachen university filtering system for the wmt 2018 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 946–954.

Julian Schamper, Jan Rosendahl, Parnia Bahar, Yunsu Kim, Arne Nix, and Hermann Ney. 2018. The rwth aachen university supervised machine translation systems for wmt 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 496–503.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. 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*, pages 223–231, Cambridge, Massachusetts, USA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

David Vilar, Daniel Stein, Matthias Huck, and Hermann Ney. 2010. Jane: Open source hierarchical translation, extended with reordering and lexicon models. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 262–270. Association for Computational Linguistics.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. Character: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 505–510.