DFKI-NMT Submission to the WMT19 News Translation Task

Jingyi Zhang[1], **Josef van Genabith**[1,2]
[1]German Research Center for Artificial Intelligence (DFKI),
Saarbrücken, Germany
[2]Department of Language Science and Technology,
Saarland University, Germany
Jingyi.Zhang@dfki.de,Josef.Van_Genabith@dfki.de

Abstract

This paper describes the DFKI-NMT submission to the WMT19 News translation task. We participated in both English-to-German and German-to-English directions. We trained standard Transformer models and adopted various techniques for effectively training our models, including data selection, back-translation, in-domain fine-tuning and model ensemble. We show that these training techniques improved the performance of our Transformer models up to 5 BLEU points. We give a detailed analysis of the performance of our system.

1 Introduction

This paper describes the DFKI-NMT submission to the WMT19 News translation task. We participated in both English-to-German and German-to-English directions. We trained Transformer models (Vaswani et al., 2017) using Sockeye[1] (Hieber et al., 2017). Compared to RNN-based translation models (Bahdanau et al., 2014), Transformer models can be trained very fast due to parallelizable self-attention networks. We applied several very useful techniques for effectively training our models.

Data Selection The parallel training data provided for German-English is quite large (38M sentence pairs). Most of the parallel data is crawled from the Internet and is not in News domain. Out-of-domain training data can hurt the translation performance on News test sets (Wang et al., 2017) and also significantly increase training time. Therefore, we trained neural language models on a large monolingual News corpus to perform data selection (Schamper et al., 2018).

Back-translation Large monolingual data in the News domain is provided for both German and English, which can be back-translated as additional parallel training data for our system (Sennrich et al., 2016a; Fadaee and Monz, 2018). The back-translated parallel data is in the News domain, which is a big advantage compared to out-of-domain parallel training data provided for the News task.

In-domain Fine-tuning The Transformer models were finally fine-tuned using the small in-domain parallel data provided for the News task (Luong and Manning, 2015; Schamper et al., 2018). Note that the large back-translated parallel data is also in-domain, but it has relatively low quality due to translation errors.

Model Ensemble We trained two Transformer models with different sizes, Transformer-base and Transformer-big. Our final submission is an ensemble of both models (Schamper et al., 2018). The ensemble of both models outperformed a single base or big model most likely because the two models can capture somewhat different features for the translation task.

2 System Details

2.1 Data Selection

The parallel data provided for the German-to-English and English-to-German tasks includes Europarl v9, ParaCrawl v3, Common Crawl corpus, News Commentary v14, Wiki Titles v1 and Document-split Rapid corpus. We also used old test sets (*newstest2008* to *newstest2017*) for training our systems. We consider News Commentary v14 and old test sets as in-domain data and the rest as out-of-domain data. Compared to the in-domain data (356k sentence pairs), the size of the out-of-domain data (38M sentence pairs) is quite large, which makes the training process relatively slow and may also hurt the translation per-

[1] https://github.com/awslabs/sockeye

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers (Day 1) pages 440–444
Florence, Italy, August 1-2, 2019. ©2019 Association for Computational Linguistics

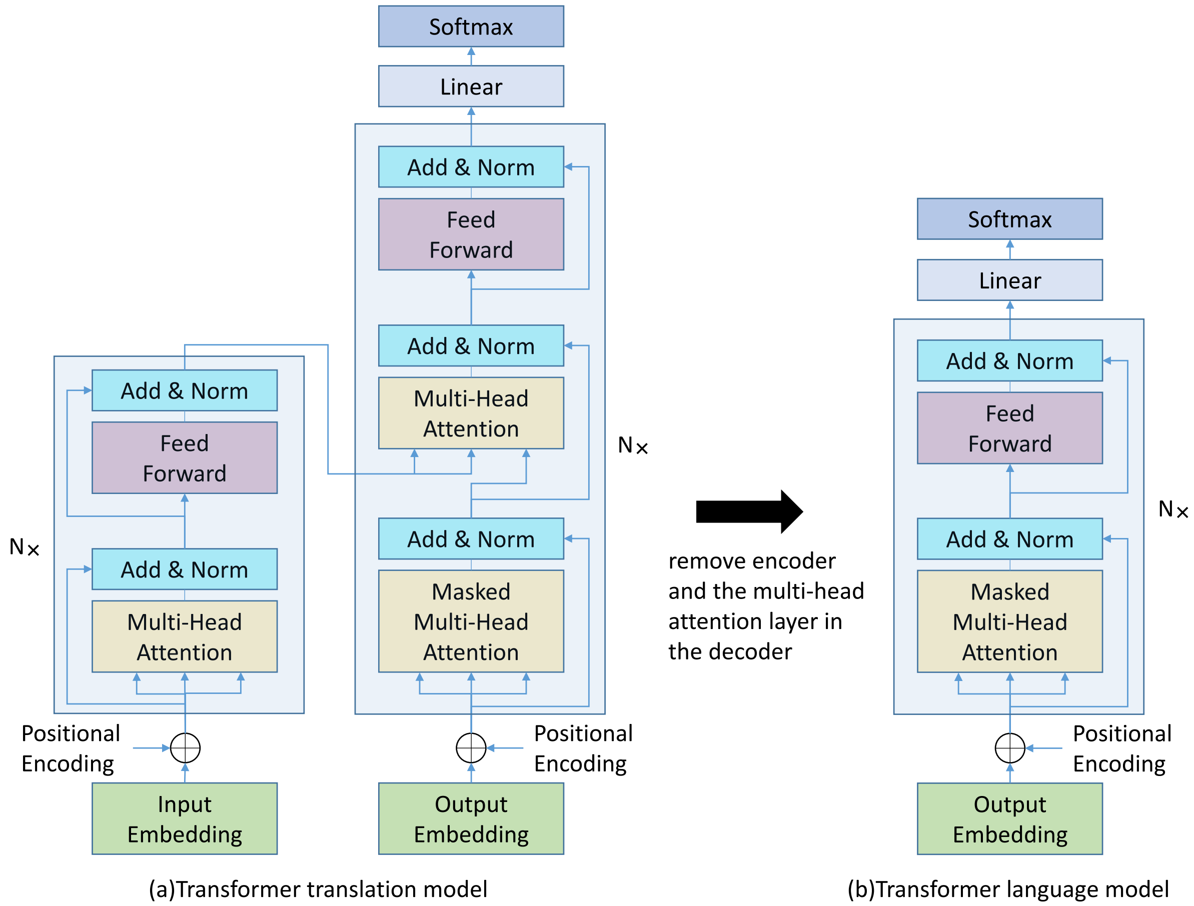

Figure 1: Structures of Transformer translation models and Transformer language models used in our experiments.

formance due to domain dismatch. Therefore, we performed data selection on out-of-domain data.

Inspired by Schamper et al. (2018)'s work which used KenLM (Heafield, 2011) for data selection, we trained two neural language models based on self-attention networks using the 2018 part of the large monolingual *News crawl* corpus for English and German, respectively. Because these neural language models are trained on the News domain, we can use them to score out-of-domain data. Sentences with higher probabilities are more likely to be in News domain. Equation 1 is used to score each sentence pair in the out-of-domain corpus. In Equation 1, P_s is the language model probability of the source sentence; N_s is the length of the source sentence; P_t is the language model probability of the target sentence; N_t is the length of the target sentence. We selected the top 15M scored sentence pairs from out-of-domain data for training our systems.

$$\frac{\log P_s}{N_s} + \frac{\log P_t}{N_t} \tag{1}$$

The neural language models trained for data selection in our experiments are based on self-attention networks which can be trained very fast. Figure 1 (a) shows the structure of the standard Transformer translation model (Vaswani et al., 2017) and we removed the encoder and the attention layer in the decoder from the Transformer translation model to create our Transformer language model as shown in Figure 1 (b). For training efficiency, we used byte pair encoding (Sennrich et al., 2016b) to learn a vocabulary of 50k for English and German respectively.

2.2 Back-translation

We back-translated the 2018 part of the large monolingual in-domain *News crawl* data as additional training data for our translation systems. Fadaee and Monz (2018) showed that it is more beneficial to back-translate sentences that contain difficult words. In our experiments, we consider words which occur less than 1000 times in the bilingual training data as difficult words. Then we randomly selected 10M sentences which contain difficult words for back-translation. The mod-

	in-domain 356k	out-of-domain 15M	back-translated 10M
Stage 1	✓	✓	
Stage 2	✓	✓	✓
Stage 3	✓		

Table 1: Training data used in different training stages.

	en-de		de-en	
	base	big	base	big
Stage 1	7.3	7.6	6.6	6.8
Stage 2	0.3	0.4	0.8	1.4
Stage 3	18.5	18.5	12.4	12.4

Table 2: Training epochs for different training stages.

els used for back-translating monolingual data are baseline Transformers (Vaswani et al., 2017) trained on the bilingual data after data selection as described before. During back-translation, we used greedy search instead of beam search for efficiency.

2.3 Model and Training

We trained two Transformer models for each translation task as Transformer-base and Transformer-big. The settings of Transformer-base is the same as the baseline Transformer in Vaswani et al. (2017)'s work. For Transformer-big, we changed word embedding size into 1024 and kept other parameters unchanged. A joint vocabulary of 50k for German and English is learned by byte pair encoding (BPE) (Sennrich et al., 2016b).[2] We set dropout to 0.1 for both Transformer-base and Transformer-big. We used adam (Kingma and Ba, 2014) for optimization. We used *newstest2018* as the validation set for model training. The training processes for both Transformer-base and Transformer-big consist of three stages.

Stage 1 We first trained the Transformers using bilingual training data, including all in-domain data and selected out-of-domain data as described in section 2.1. Note that the back-translated data was not used in this stage. Each training batch contains 8192 words and the validation frequency is 2000 batches. We set the initial learning rate to be 2.00e-04. We reduced the learning rate by a factor of 0.70 whenever the validation score does not improve 8 times. We stopped the training process after 5 times of learning rate reduction.

	en-de		de-en	
	base	big	base	big
Stage 1	44.24	45.03	45.34	45.75
Stage 2	46.42	47.10	47.84	48.65
Stage 3	47.80	48.83	48.65	49.33
Ensemble	49.45		49.75	

Table 3: Case-insensitive BLEU scores on *newstest2018*. "Ensemble" means ensemble both Transformer-base and Transformer-big after Stage 3.

Stage 2 We used all bilingual training data used in the first training stage together with the back-translated monolingual data to continue training the models which had converged in the first training stage. We kept the batch size to be 8192 words and changed the validation frequency to 1000 batches. We set the initial learning rate to be 1.00e-05 and stopped the training process when the validation score does not improve 8 times.

Stage 3 For fine-tuning, we used the small parallel in-domain data as described in section 2.1 to continue training the models which had converged in the second training stage. We changed batch size to be 1024 words and validation frequency to be 100 batches. We set the initial learning rate to be 1.00e-06 and stopped the training process when the validation score does not improve 8 times.

Table 1 shows training data used in different training stages. The models trained in the first training stage were used to back-translate monolingual data as described in section 2.2. In Stage 2, we continued training the models which had converged in Stage 1 instead of training models with random initialization in order to reduce the training time of Stage 2.

2.4 Results and Analysis

Table 2 shows the numbers of training epochs for different training stages and Table 3 shows the performance of our systems after different training stages. As we can see, back-translation (Stage 2) and in-domain fine-tuning (Stage 3) both improved the translation quality on a significant level. An ensemble of Stage 3 Transformer-base and Transformer-big achieved further improvements. We also tried to ensemble different checkpoints of Transformer-big, but achieved little improvement, likely because different checkpoints of

[2] For preprocessing, we used Moses (Koehn et al., 2007) scripts *normalize-punctuation.perl*, *tokenizer.perl*, *lowercase.perl*. We trained a recaser using *train-recaser.perl* to recase translations.

Example 1	
Src	wei@@ dez@@ aun@@ projekt ist element@@ ar
Ref	past@@ ure fence project is fundamental
Ours	electric sound project is elementary
Example 2	
Src	jetzt nimmt sich das weiße haus von trump die freiheits@@ statue vor
Ref	now trump &apos;s white house is targeting the statue of liberty
Ours	now trump &apos;s white house takes the statue of liberty

Table 4: Translation examples. "@@" means segmented by *byte pair encoding*.

the same model are very similar.

In addition, we give some translation examples in Table 4 to analyze when and why our translation system makes mistakes. The translations in Table 4 are produced by our best system, i.e., ensemble of Transformer-base and Transformer-big after training stage 3. In Example 1, "wei@@ dez@@ aun@@ projekt" (pasture fence project) is wrongly translated into "electric sound project", likely because "weidezaunprojekt" is a unknown word and does not occur in the training data. Although BPE can help to relieve data sparsity by using smaller and more frequent sub-word units, the automatic BPE segmentation "wei@@ dez@@ aun@@ projekt" is a bad segmentation with linguistically meaningless sub-word pieces. A better segmentation "weide@@(pasture) zaun@@(fence) projekt" may help to reduce data sparsity and get better translation. Example 2 does not contain rare words, but "nimmt vor" is still wrongly translated into "takes". This is likely because "nimmt vor" has different translations in the training data and the correct translation here "targeting" is relatively uncommon. We find many translation mistakes of our system are caused by rare words or uncommon usages of words as shown in Table 4, which we will work on in the future.

3 Conclusion

This paper describes the DFKI-NMT submission to the WMT19 English-to-German and German-to-English News translation tasks. We trained standard Transformer models and adopted various techniques for effectively training our models, including data selection, back-translation, in-domain fine-tuning and model ensemble. We show that effective training techniques can improve the performance of standard Transformer models up to 5 BLEU points.

Acknowledgments

This work is supported by the German Federal Ministry of Education and Research (BMBF) under the funding code 01IW17001 (Deeplee). Responsibility for the content of this publication is with the authors.

References

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

Marzieh Fadaee and Christof Monz. 2018. Back-translation sampling by targeting difficult words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 436–446.

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

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*.

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

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 Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the In-*

ternational Workshop on Spoken Language Translation, pages 76–79.

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. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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)*, pages 1715–1725.

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.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 560–566.