The Karlsruhe Institute of Technology Systems for the News Translation Task in WMT 2018

Ngoc-Quan Pham and **Jan Niehues** and **Alex Waibel**
Karlsruhe Institute of Technology
ngoc.pham@kit.edu jan.niehues@kit.edu alex.waibel@kit.edu

Abstract

We present our experiments in the scope of the news translation task in WMT 2018, in directions: English→German. The core of our systems is the encoder-decoder based neural machine translation models using the transformer architecture. We enhanced the model with a deeper architecture. By using techniques to limit the memory consumption, we were able to train models that are 4 times larger on one GPU and improve the performance by 1.2 BLEU points. Furthermore, we performed sentence selection for the newly available ParaCrawl corpus. Thereby, we could improve the effectiveness of the corpus by 0.5 BLEU points.

1 Introduction

This manuscript provides the technical details regarding our submission in the WMT18 shared task on English→German news translation. Our submission has two major research contributions: Firstly, the development of a deep, efficient neural architectures and secondly, the cleaning and data selection of web crawled data.

We developed a efficient approach to train a deep transformer model on a single GPU. This allows use to train a 4 times deeper model than state-of-the-art models on one GPUs. In the experiments we are able to show that these models perform 1.2 BLEU points better than the baseline model using already 8 layers.

Secondly, we performed additional filtering on the ParaCrawl corpus. We are using the log-probabilities of a baseline NMT system to filter the low quality translations. While we are only able to improve the translation quality slightly by 0.3 BLUE points using all ParaCrawl data, the integration of the clean cropus improved the translation quality of 0.8 BLEU points.

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 English↔German

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, the ParaCrawl corpus and simulated data. The preprocessing includes tokenization, removing very long sentences and the sentence pairs which are length-mismatched, normalizing special symbols and different writing rules and smart-casing the first word of each sentence. Those tools are provided in the Moses Toolkit [1].

We integrated the monolingual news data by generating synthetic data as motivated by Sennrich et al. (2016a). We used the translated data provided by University of Edinburgh.

Once the data is preprocessed, we applied byte-pair encoding (BPE) (Sennrich et al., 2016b) on the corpus. In this work, we deploy an operation size of 40K (shared between English and German languages) and applied vocabulary filtering in a way that every token occurs at least 50 times.

2.1.1 ParaCrawl data selection

This year, in addition to the data provided in the last years, also the ParaCrawl corpus was provided. Since this data is collected by a web-crawler it differs in several ways from the other corpus. Firstly, it is significantly larger than all other available corpora. But the corpus is also more noisy. Therefore, we did not directly use this corpus, but filtered it prior to training.

[1] http://www.statmt.org/moses/

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 471–476

https://doi.org/10.18653/v1/W18-6422

In this use case, an NMT system trained on the clean parallel data was evaluated. Therefore, we investigate the usage of this system to select clean translations from the training data.

In a first step, we performed the same preprocessing as for the other corpora. In addition, we removed short sentences. We noticed that these were often only keywords or numbers and therefore would not be helpful to train the system. In our experiments, we removed all sentences shorter than $n = 10$ words.

In the second step, we use the NMT system to calculate the translation probability of the ParaCrawl data. We used the length normalized log probability to select the sentences used for training. An inspection on a tiny subset of the data showed that the sentences with a low length-normalized probability seem to be bad translations. Examples are shown in the top of Table 1. Often the they are even not sentences in the source and target language. Furthermore, we noticed that the sentences with a very high probability seem not to be very useful. As shown in the last example in 1, in these cases, we often have a one-to-one word correspondence between the source and target sentence. But the input are often no real sentences and therefore, we might learn to generate no longer fluent output.

Due to computation time, we were not able to train model on different selected parts of the corpus. In contrast, we select reasonable thresholds based on the ordering on a small subset. We removed all sentences, where the length-normalized log-probability is smaller then $a = 0.8$ and all sentences where this score is higher than $b = 3.6$.

3 Deep Transformer

The research in Machine Translation has observed rapid advancement in terms of modeling in the past three years. While recurrent neural networks remain the core component in many strong systems (Wu et al., 2016), various works incrementally discovered that other architectures can also outperform RNNs in terms of translation quality or training efficiency, such as Convolutional Neural Networks (CNNs) (Gehring et al., 2017) or Self-Attention Networks, or Transformer (Vaswani et al., 2017). Due to the success of the self-attention networks, we will concentrate in this work on this type of architecture.

While other areas of deep learning use very deep neural networks, the networks used for NMT are still shallow compared to these areas. Motivated by the success of deep models in other areas, we analyzed the effectiveness of depth of the Transformer network. This is only possible trough the development of a very efficient implementation. This enables us to training very deep networks on a single device in a reasonable amount of time.

3.1 Sequence-to-Sequence models

Neural machine translation (NMT) consists of an encoder and a decoder (Sutskever et al., 2014; Cho et al., 2014) that directly approximate the conditional probability of a target sequence $Y = y_1, y_2, \cdots, y_T$ given a source sequence $X = x_1, x_2, \cdots, x_M$. The basic concept of the model is to encode the source sequence with a neural network to capture the neural representation of the source sentence, which is then referred multiple times during a decoding process, in which another neural network auto-regressively generates tokens in the target language.

The architectural choice is important in building neural machine translation systems. While Recurrent Neural Networks (RNN) have become the defacto model to represent sequences and were applied very successfully in NMT (Sutskever et al., 2014; Luong and Manning, 2015), self-attention networks (or Transformer) arose as a potentially better alternative (Vaswani et al., 2017).

3.2 Transformer overview

The transformer architecture was previously introduced with the following novel features:

- Long range dependency is modeled using the self-attention mechanism instead of recurrent connections used in recurrent networks, like the Long-Short Term Memories. The mechanism allows direct connection between two different two arbitrary positions in the sequences, which in turns alleviates the gradient flow problem existing in recurrent networks.

- Residual block design: similar to the infamous residual networks consisting of deep convolutional neural networks, Transformer networks are built on residual blocks in which the lower level states are directly carried to the top level by addiction. In the

German:	offener Teilnahmewettbewerb : Grafikdesign fr Musikprojekt
English:	DAS Hotel
German:	anderen Gewinnen .
English:	Anyway , I will repeat that I sincerely hope you weren 't referring to me
German:	Christijan Albers 2 : 2 (3 : 2 im Elfmeterschieen)
English:	Christijan Albers 2 : 2 (3 : 2 in penalty shootout)

Table 1: Filtered examples

Transformer networks, the input of every sub-block is added directly to the output (He et al., 2016), as a result the final layer receives a large sum of inputs from below, including the embeddings.

- Multi-head attention being proposed as a variation of the attention network (Bahdanau et al., 2014) improves attention power by performing attention in multiple dimensions of the input, which are projected using linear transformation.

- Additional neural network training utilities: layer normalization (Ba et al., 2016) prevents network state values from exploding; label smoothing regularizes the cross entropy loss function to improves the models' generalization;

3.3 Efficient memory usage

NMT models in general are very memory consuming due to the fact that they need to apply transformation on a sequence of states instead of single states in feed-forward neural networks. For other architectures, like feed-forward neural networks, convolution neural networks and recurrent neural networks, recently techniques have been proposed to significantly reduce the memory footprint during training (Chen et al., 2016; Gruslys et al., 2016). The main idea is to recalculate intermediate results instead of caching them. In this work, we adopted this idea to transformer models. We apply the method for a layer basis, by specifying the number of layers (Transformer Encoder or Decoder block) to be checkpointed during training. Such layer's forward pass needs to be recomputed during the backward pass, as a result the intermediate buffers created during training can be discarded, resulting in smaller memory requirement and bigger batch size.

3.4 Training

We followed the original work for the general hyper parameters including batch size and learning rate. We instead focus on several methods to increase training efficiency of the Transformer models.

Emulated Multi-GPU setup: It is notable that the *Noam* learning rate schedule proposed in (Vaswani et al., 2017) was designed for bigger batch sizes (≈ 25000 words per mini-batch update which is not feasible for a single-GPU setup). In order to apply the same learning schedule without a multi-GPU system, we simply divide the large mini-batch into smaller ones, and accumulate (by summing) the gradients computed by each mini-batch forward and backward pass.

4 Results

4.1 Baseline System

Our baseline system uses the openNMT-py Toolkit[2] and uses an RNN based translation model with 4 layers in both decoders and encoders (bidirectional RNN on the encoder side). The model is equipped with dropout$= 0.2$ following the work of (Zaremba et al., 2014) for better regularization and label smoothing improving the cross-entropy loss. The training details and hyper-parameters are replicated from (Pham, 2017). In all of our experiments, we use the concatenation of test sets from 2013 to 2016 as our development set for model/checkpoint selection. While we use perplexity for model selection, the BLEU score on newstest2017 calculated by mteval-v13a.pl is used to report the models' performance.

4.2 Training hyper parameters

For RNN models, we use 4-layer-models with Long-Short Term Memory (Hochreiter and Schmidhuber, 1997). The bi-directional LSTM is used in the Encoder for all 4 layers. We use batch

[2]http://opennmt.net

size of 128 sentences (notably, the measurement of batch size in Transformer is denoted by the number of tokens, not sentences) and simply trained with Stochastic Gradient Descent with learning rate decay when the validation perplexity does not improve (Luong et al., 2015).

For Transformer models, we set the base layer size to 512, while the hidden layer in each Position Wise Feed Forward network has 2048 neurons, which matches the *Base* model in (Vaswani et al., 2017).

The learning method is Adam (Kingma and Ba, 2014) with the learning rate schedule similar to the original paper, with a minor difference that we increase the number of warm up steps to 8192 and double the base learning rate. If Dropout is applied, we use dropout at each Position Wise Feed Forward hidden layer and the attention weights.

4.3 Model comparison

In a first series of experiment we compared different architectures (RNNs and Transformers) and the influence of the deeps of the network. The transformer-based models are implemented using PyTorch (Paszke et al., 2017) and the source codes are open sourced. [3]. We provided our starting point as a reference to our participation to the last year's shared task. Thus, we use the corpus consisting of the Europarl, News Commentary, Rapid Corpus and the cleaned Common Crawl, which is then boosted with the back translation data provided by University of Edinburgh. The total data size is around 9 million sentence pairs.

Model	BLEU (newstest2017)
Baseline (RNN)	27.4
Transformer-4	27.8
Transformer-12	29.2
Transformer-24	29.7

Table 2: RNNs vs Transformers (various depths) trained without paraCrawl.

As the results in Table 2 suggest, the baseline model despite having larger model size (1024) and being improved with dropout and label smoothing is not able to outperform a base Transformer (hidden size 512 for every layer) with only 4 layers. More importantly, the result scales over the Transformer's depth, such subject will be covered in the subsequent section. We managed to outperform the RNN baseline by 2.3 BLEU points just by increasing the depth to 24 layers.

Though we do not provide any comparison with respect to depth in Recurrent Neural Networks, previous work (Britz et al., 2017) explores different depths during training NMT models with similar architectures to our baseline discovering that it is not trivial to improve Recurrent NMT models just by increasing depth even with residual connections. It is notable that recent work (Chen et al., 2018) empirically proved that RNN models with hyper parameter tuning and layer normalization strategy can perform on par with the Transformer.

4.4 Data Size

As illustrated above, the Transformer models produced strong results which can outperform the best system of last year which is an ensemble of RNN models (Sennrich et al., 2017). We proceed to improve the system further by providing additional training data. Table 3 shows that a naive addiction of the paraCrawl data yields only a boost of 0.3 BLEU points, while our filtering method impressively improves the result by 0.8.

Data	News2017
Transformer-12	29.2
+paraCrawl	29.5
+ filtered paraCrawl	30.0

Table 3: Experiments using different data sizes

4.5 When do we need regularization

Deeper models are more likely to overfit, which can be alleviated by using Dropout, specifically in the Position-wise feed forward network in each transformer block. We apply dropout at the the embeddings, residual connections (the output of the transformations before addiction) and at the attention matrices with the same probability of 0.1) The results in table 4 shows that Dropout started to be effective when the model becomes deeper than 12, even though the difference in the 16 configuration is rather subtle. At 12 layers and below, dropout seems to be unnecessary, possibly because our corpus size has reach 40 million sentences (included the filtered paraCrawl corpus).

Since we used the training regime which stops after $100K$ steps (each updates the parameter based on the batch size of about 25000 words), it is possible that Dropout models requires training

[3] https://github.com/isl-mt/NMTGMinor

for more than such threshold, due to the fact that a side effect of Dropout is to prolong the training progress.

4.6 Deeper networks

To answer the empirical question if very deep networks can improve the translation performance given abundant training data (as we have three times more data than the first experiment w.r.t depth), we managed to train networks as deep as 32 layers. The results are shown in Table 5. We observe significance improvement (0.7 BLEU points) in the first incremental steps from 4 to 12. The progress becomes stagnant from 16, and not until reaching 32 layers did we manage to obtain an additional 0.4 increase. The Transformer network clearly benefits from depth, which was not observed in Recurrent Network (Britz et al., 2017), however the effect is diminishing at 12 layers, while training models as deep as 32 is not simple. To the best of our knowledge, our model consists of totally 96+48+2 sub-blocks (encoders, decoders and input/output layers) which is the first attempt to explore a network with this depth in Neural Machine Translation.

Our training time ranges from 1 week with the 12-layer models to maximum of 2 weeks for the 32-layer models using single GTX 1080Ti graphics cards.

4.7 Final submission

The final submission of KIT is the ensemble of 5 models using different layer sizes and switching on and of dropout. Each of the models is already an average of different checkpoints. The results are summarized in Table 6. We found that the an ensemble of 5 models is only able to increase the score by 0.3, which shows that the 32-layer model dominates others.

5 Conclusion

In conclusion, we described our experiments in the news translation task in WMT 2018. The main focus of our submission was on data selection and techniques to efficient train deep transformer models . While we were only able to improve the translation performance slightly by using the whole ParaCrawl corpus, we could improve the translation performance by 0.8 BLEU points when using a filtered version of the corpus. We successfully filtered the data by using the translation probabilities of a baseline NMT system. Secondly, we were successfully in training a deep transformer model on a single GPU. By increasing the depth of the network by a factor of 4, we were able to gain additional 1.2 BLEU points. This was only possible by caching less data during training and recalculating them if needed.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

D. Bahdanau, K. Cho, and Y. Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849*.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia. PMLR.

Audrunas Gruslys, Rémi Munos, Ivo Danihelka, Marc Lanctot, and Alex Graves. 2016. Memory-efficient backpropagation through time. In *Advances in Neural Information Processing Systems*, pages 4125–4133.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

	No Dropout		Dropout	
Layers	dev(ppl)	News2017	dev(ppl)	News2017
12	3.5	30.0	3.45	29.7
16	3.6	29.7	3.41	29.8

Table 4: Experiments using dropout, with large data including filtered paraCrawl.

Layer	News20(13-16) (ppl)	News2017 (BLEU)
4	4.0	28.5
8	3.7	29.2
12	3.5	30.0
16	3.4	29.8
32	**3.2**	**30.4**

Table 5: Experiments using different layers, with large data including filtered paraCrawl.

Layer	News2017
12-layer	30.0
12-layer dropout	29.7
16-layer	29.7
16-layer dropout	29.8
32-layer	30.4
ensemble	**30.7**

Table 6: Systems used in the final submission

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

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

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

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

N-Q Pham. 2017. The karlsruhe institute of technology systems for the news translation task in wmt 2017. In *Proceedings of the Second Conference on Statistical Machine Translation (WMT 2017)*, Copenhagen, Dennmark.

R. Sennrich, B. Haddow, and A. Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54st Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

R. Sennrich, B. Haddow, and A. Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54st Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112, Quebec, Canada.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.