UdS Submission for the WMT 19 Automatic Post-Editing Task

Hongfei Xu
Saarland University
DFKI
hfxunlp@foxmail.com

Qiuhui Liu
China Mobile Online Services
liuqiuhui@cmos.chinamobile.com

Josef van Genabith
Saarland University
DFKI
josef.van_genabith@dfki.de

Abstract

In this paper, we describe our submission to the English-German APE shared task at WMT 2019. We utilize and adapt an NMT architecture originally developed for exploiting context information to APE, implement this in our own transformer model and explore joint training of the APE task with a de-noising encoder.

1 Introduction

The Automatic Post-Editing (APE) task is to automatically correct errors in machine translation outputs. This paper describes our submission to the English-German APE shared task at WMT 2019. Based on recent research on the APE task (Junczys-Dowmunt and Grundkiewicz, 2018) and an architecture for the utilization of document-level context information in neural machine translation (Zhang et al., 2018b), we re-implement a multi-source transformer model for the task. Inspired by Cheng et al. (2018), we try to train a more robust model by introducing a multi-task learning approach which jointly trains APE with a de-noising encoder.

We made use of the artificial eScape data set (Negri et al., 2018) provided for the task, since the multi-source transformer model contains a large number of parameters and training with large amounts of supplementary synthetic data can help regularize its parameters and make the model more general. We then tested the BLEU scores between machine translation results and corresponding gold standard post-editing results on the original development set, the training set and the synthetic data as shown in Table 1.

dev	train	eScape
77.15	77.42	37.68

Table 1: BLEU Scores of Data Sets

Table 1 shows that there is a significant gap between the synthetic eScape data set (Negri et al., 2018) and the real-life data sets (the development set and the original training set from post-editors), potentially because Negri et al. (2018) generated the data set in a different way compared to Junczys-Dowmunt and Grundkiewicz (2016) and very few post-editing actions are normally required due to the good translation quality of neural machine translation (Bahdanau et al., 2014; Gehring et al., 2017; Vaswani et al., 2017) which significantly reduces errors in machine translation results and makes the post-editing results quite similar to raw machine translation outputs.

2 Our Approach

We simplify and employ a multi-source transformer model (Zhang et al., 2018b) for the APE task, and try to train a more robust model through multi-task learning.

2.1 Our Model

The transformer-based model proposed by Zhang et al. (2018b) for utilizing document-level context information in neural machine translation has two source inputs which can also be a source sentence along with the corresponding machine translation output and therefore caters for the requirements of APE. Since both source sentence and machine translation outputs are important for the APE task (Pal et al., 2016; Vu and Haffari, 2018), we remove the context gate used to restrict the information flow from the first input to the final output in their architecture, and obtain the model we used for our submission shown in Figure 1.

The model first encodes the given source sentence with stacked self-attention layers, then "post-edits" the corresponding machine translation result through repetitively encoding the machine translation result (with a self-attention

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 3: Shared Task Papers (Day 2) pages 147–152
Florence, Italy, August 1-2, 2019.

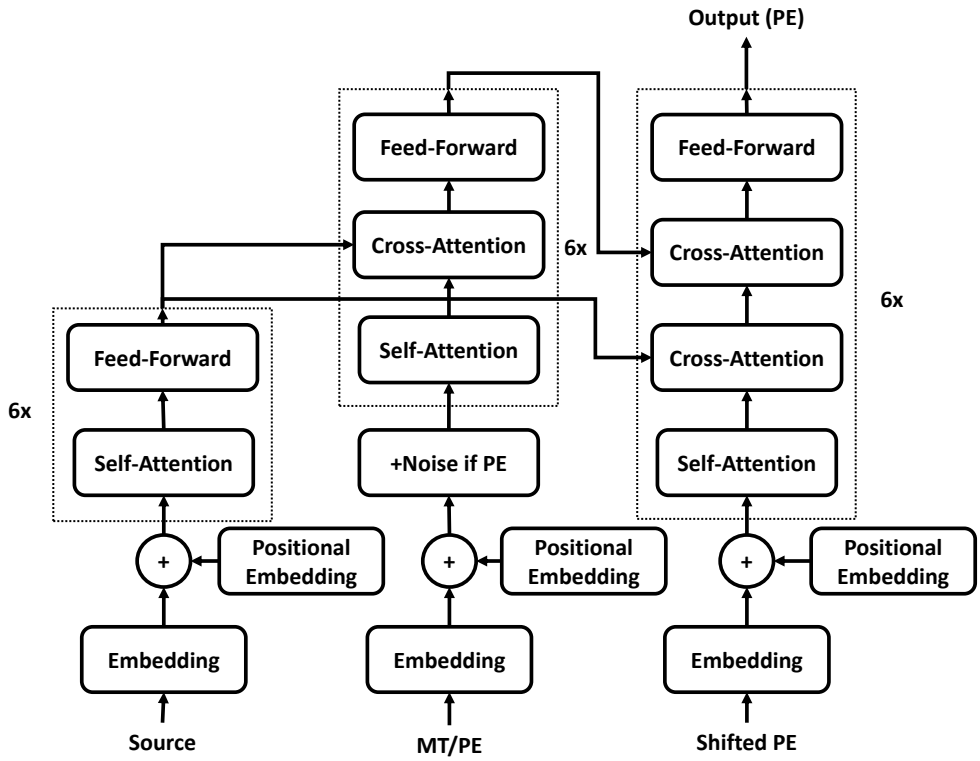

Figure 1: Our Transformer-Based Multi-Source Model for the APE Task

layer), attending to the source sentence (with a cross-attention layer) and processing the collected information (with a feed-forward neural network). Finally, the decoder attends to representations of the source sentence and the machine translation result and generates the post-editing result.

Compared to the multi-source transformer model used by Junczys-Dowmunt and Grundkiewicz (2018), this architecture has one more cross-attention module in the encoder for machine translation outputs to attend to the source input which makes the parameter sharing of layers between two encoders impossible, but we think this cross-attention module can help the de-noising task. The embedding of source, machine translation outputs and post-editing results is still shared as Junczys-Dowmunt and Grundkiewicz (2018) advised.

2.2 Joint Training with De-noising Encoder

Table 1 shows a considerable difference between the synthetic data set (Negri et al., 2018) and the real data set. To enable the model to handle more kinds of errors, we simulate new "machine translation outputs" through adding noise to the corresponding post-editing results. Following Cheng et al. (2018), we add noise directly to the look-up embedding of post-editing results instead of manipulating post-editing sequences.

Since the transformer (Vaswani et al., 2017) does not apply any weight regularization, we assume that the model can easily learn to reduce noise by enlarging weights, and propose to add adaptive noise to the embedding:

$$emb_{out} = emb + strength * \overline{abs(emb)} * N \quad (1)$$

where emb is the embedding matrix, $strength$ is a number between $[0.0, +\infty)$ to control the strength of noise, N is the noise matrix of the same shape as emb. We explore both standard Gaussian distribution and uniform distribution of $[-1.0, -1.0]$ as N. In this way the noise will automatically grow with the growing embedding weights.

Given that the transformer translation model (Vaswani et al., 2017) incorporates word order information through adding positional embedding to word embedding, we add noise to the combined embedding. In this case, the noise can both affect the word embedding (replacing words with their synonyms) and positional embedding (swapping word orders).

During training, we use the same model, and achieve joint training by randomly varying inputs: the inputs for the APE task are {source, mt, pe},

while those for the de-noising encoder task are {source, pe+noise, pe} where "source", "mt" and "pe" stand for the source sentence, the corresponding output from the machine translation system and the correct post-editing result. The final loss for joint training is:

$$loss = \lambda * loss_{\text{ape}} + (1 - \lambda) * loss_{\text{de-noising}} \quad (2)$$

i.e. the loss between the APE task and the de-noising encoder task are balanced by λ in this way.

3 Experiments

We implemented our approaches based on the Neutron implementation (Xu and Liu, 2019) for transformer-based neural machine translation.

3.1 Data and Settings

We only participated in the English to German task, and we used both the training set provided by WMT and the synthetic eSCAPE corpus (Negri et al., 2018). We first re-tokenized[1] and truecased both data sets with tools provided by Moses (Koehn et al., 2007), then cleaned the data sets with scripts ported from the Neutron implementation, and the original training set was up-sampled 20 times as in (Junczys-Dowmunt and Grundkiewicz, 2018). We applied joint Byte-Pair Encoding (Sennrich et al., 2016) with $40k$ merge operations and 50 as the vocabulary threshold for the BPE. We only kept sentences with a max of 256 sub-word tokens for training, and obtained a training set of about $6.5M$ triples with a shared vocabulary of 42476. We did not apply any domain adaptation approach for our submission considering that (Junczys-Dowmunt and Grundkiewicz, 2018) shows few improvements, but advanced domain adaption (Wang et al., 2017) or fine-tuning (Luong and Manning, 2015) methods may still bring some improvements. The training set was shuffled for each training epoch.

Like Junczys-Dowmunt and Grundkiewicz (2018), all embedding matrices were bound with the weight of the classifier. But for tokens which in fact do never appear in post-editing outputs in the shared vocabulary, we additionally remove their weights in the label smoothing loss and set corresponding biases in the decoder classifier to -10^{32}.

Unlike Zhang et al. (2018b), the source encoder, the machine translation encoder and the decoder had 6 layers. The hidden dimension of the position-wise feed-forward neural network was 2048, the embedding dimension and the multi-head attention dimension were 512. We used a dropout probability of 0.1, and employed label smoothing (Szegedy et al., 2016) value of 0.1. We used the Adam optimizer (Kingma and Ba, 2015) with 0.9, 0.98 and 10^{-9} as β_1, β_2 and ϵ. The learning rate schedule from Vaswani et al. (2017) with $8,000$ as the number of warm-up steps[2] was applied. We trained our models for only 8 epochs with at least $25k$ post-editing tokens in a batch, since we observed over-fitting afterwards. For the other hyper parameters, we used the same as the transformer base model (Vaswani et al., 2017).

During training, we kept the last 20 checkpoints saved with an interval of $1,500$ training steps (Vaswani et al., 2017; Zhang et al., 2018a), and obtained 4 models for each run through averaging every 5 adjacent checkpoints.

For joint training, we simply used 0.2 as the strength of noise ($strength$), and 0.5 as λ for joint training. Other values may provide better performance, but we did not have sufficient time to try this for our submission.

During decoding, we used a beam size of 4 without any length penalty.

3.2 Results

We first evaluated case-sensitive BLEU scores[3] on the development set, and results of all our approaches and baselines are shown in Table 2.

"MT as PE" is the do-nothing baseline which takes the machine translation outputs directly as post-editing results. "Processed MT" is the machine translation outputs through pre-processing (re-tokenizing and truecasing) and post-processing (de-truecasing and re-tokenizing without "-a" argument[4]) but without APE. "Base", "Gaussian" and "Uniform" stand for our model trained only for the APE task, jointly trained with Gaussian noise and uniform noise, respectively. We reported the minimum and the maximum BLEU scores of the 4 averaged models for

[1] using arguments: -a -no-escape

[2] https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/transformer.py#L1623.

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl.

[4] "-a" indicates tokenizing in the aggressive mode, which normally helps reduce vocabulary size. The official data sets were tokenized without this argument, so we have to recover our post-editing outputs.

each experiment. "Ensemble x5" is the ensemble of 5 models from joint training, 4 of which were averaged models with highest BLEU scores on the development set, another one was the model saved for each training epoch with lowest validation perplexity.

Models	BLEU
MT as PE	76.76
Processed MT	76.61
Base	76.91 ∼ 77.13
Gaussian	76.94 ∼ 77.08
Uniform	77.01 ∼ 77.10
Ensemble x5	**77.22**

Table 2: BLEU Scores on the Development Set

Table 2 shows that the performance got slightly hurt (comparing "Processed MT" with "MT as PE") with pre-processing and post-processing procedures which are normally applied in training seq2seq models for reducing vocabulary size. The multi-source transformer (Base) model achieved the highest single model BLEU score without joint training with the de-noising encoder task. We think this is perhaps because there is a gap between the generated machine translation outputs with noise and the real world machine translation outputs, which biased the training.

Even with the ensembled model, our APE approach does not significantly improve machine translation outputs measured in BLEU (+0.46). We think human post-editing results may contain valuable information to guide neural machine translation models in some way like Reinforcement-Learning, but unfortunately, due to the high quality of the original neural machine translation output, only a small part of the real training data in the APE task are actually corrections from post editors, and most data are generated from the neural machine translation system, which makes it like adversarial training of neural machine translation (Yang et al., 2018) or multi-pass decoding (Geng et al., 2018).

All our submissions were made by jointly trained models because the performance gap between the best and the worst model of jointly trained models is smaller, which means that jointly trained models may have smaller variance.

Results on the test set from the APE shared task organizers are shown in Table 3. Even the ensemble of 5 models did not result in significant differences especially in BLEU scores.

Models	TER	BLEU
MT as PE	16.84	74.73
Gaussian	16.79	75.03
Uniform	16.80	75.03
Ensemble x5	16.77	75.03

Table 3: Results on the Test Set

4 Related Work

Pal et al. (2016) applied a multi-source sequence-to-sequence neural model for APE, and Vu and Haffari (2018) jointly trained machine translation with the post editing sequence prediction task (Berard et al., 2017). Though all previous approaches get significant improvements over Statistical Machine Translation outputs, benefits with APE on top of Neural Machine Translation outputs are not very significant (Chatterjee et al., 2018).

On the other hand, advanced neural machine translation approaches may also improve the APE task, such as: combining advances of the recurrent decoder (Chen et al., 2018), the Evolved Transformer architecture (So et al., 2019), Layer Aggregation (Dou et al., 2018) and Dynamic Convolution structures (Wu et al., 2019).

5 Conclusion

In this paper, we described details of our approaches for our submission to the WMT 19 APE task. We borrowed a multi-source transformer model from the context-dependent machine translation task and applied joint training with a de-noising encoder task for our submission.

Acknowledgments

Hongfei Xu is supported by a doctoral grant from China Scholarship Council ([2018]3101, 201807040056). This work is supported by the German Federal Ministry of Education and Research (BMBF) under the funding code 01IW17001 (Deeplee). We thank the anonymous reviewers for their instructive comments.

References

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

Alexandre Berard, Laurent Besacier, and Olivier Pietquin. 2017. LIG-CRIStAL submission for the wmt 2017 automatic post-editing task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 623–629, Copenhagen, Denmark. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 shared task on automatic post-editing. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 723–738, Belgium, Brussels. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 76–86, Melbourne, Australia. Association for Computational Linguistics.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766, Melbourne, Australia. Association for Computational Linguistics.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4253–4262. Association for Computational Linguistics.

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.

Xinwei Geng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. Adaptive multi-pass decoder for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 523–532, Brussels, Belgium. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. MS-UEdin submission to the wmt2018 ape shared task: Dual-source transformer for automatic post-editing. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 835–839, Belgium, Brussels. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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, Prague, Czech Republic. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. ESCAPE: a Large-scale Synthetic Corpus for Automatic Post-Editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, and Josef van Genabith. 2016. A neural network based approach to automatic post-editing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 281–286, Berlin, Germany. Association for Computational Linguistics.

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)*, pages 1715–1725. Association for Computational Linguistics.

David R. So, Chen Liang, and Quoc V. Le. 2019. The evolved transformer. *CoRR*, abs/1901.11117.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Thuy-Trang Vu and Gholamreza Haffari. 2018. Automatic post-editing of machine translation: A neural programmer-interpreter approach. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3048–3053, Brussels, Belgium. Association for Computational Linguistics.

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, Vancouver, Canada. Association for Computational Linguistics.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Hongfei Xu and Qiuhui Liu. 2019. Neutron: An Implementation of the Transformer Translation Model and its Variants. *arXiv preprint arXiv:1903.07402*.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Improving neural machine translation with conditional sequence generative adversarial nets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1346–1355, New Orleans, Louisiana. Association for Computational Linguistics.

Biao Zhang, Deyi Xiong, and jinsong su jinsong. 2018a. Accelerating neural transformer via an average attention network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798. Association for Computational Linguistics.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018b. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium. Association for Computational Linguistics.