USAAR-DFKI – The Transference Architecture for English–German Automatic Post-Editing

Santanu Pal[1,2], Hongfei Xu[1,2], Nico Herbig[2], Antonio Krüger[2], Josef van Genabith[1,2]
[1]Department of Language Science and Technology,
Saarland University, Germany
[2]German Research Center for Artificial Intelligence (DFKI),
Saarland Informatics Campus, Germany
{santanu.pal, josef.vangenabith}@uni-saarland.de
{hongfei.xu, nico.herbig, krueger, josef.van_genabith}@dfki.de

Abstract

In this paper we present an English–German Automatic Post-Editing (APE) system called *transference*, submitted to the APE Task organized at WMT 2019. Our *transference* model is based on a multi-encoder transformer architecture. Unlike previous approaches, it (i) uses a transformer encoder block for src, (ii) followed by a transformer decoder block, but without masking, for self-attention on mt, which effectively acts as second encoder combining $src \rightarrow mt$, and (iii) feeds this representation into a final decoder block generating pe. This model improves over the raw black-box neural machine translation system by 0.9 and 1.0 absolute BLEU points on the WMT 2019 APE development and test set. Our submission ranked 3rd, however compared to the two top systems, performance differences are not statistically significant.

1 Introduction & Related Work

Automatic post-editing (APE) is a method that aims to automatically correct errors made by machine translation (MT) systems before performing actual human post-editing (PE) (Knight and Chander, 1994), thereby reducing the translators' workload and increasing productivity (Pal et al., 2016a; Parra Escartín and Arcedillo, 2015b,a; Pal et al., 2016a). Recent advances in APE research are directed towards neural APE based on neural MT where APE systems can be viewed as a 2nd-stage MT system, translating predictable error patterns in MT output to their corresponding corrections. APE training data minimally involves MT output (mt) and the human post-edited (pe) version of mt, but additionally using the source (src) has been shown to provide further benefits (Bojar et al., 2015, 2016, 2017). Based on the training process, APE systems can be categorized as either single-source ($mt \rightarrow pe$) or multi-source ($\{src, mt\} \rightarrow pe$) approaches. This integration of source-language information in APE is intuitively useful in conveying context information to improve APE performance. Neural APE was first proposed by Pal et al. (2016b) and Junczys-Dowmunt and Grundkiewicz (2016). A multi-source neural APE system can be configured either by using a single encoder that encodes the concatenation of src and mt (Niehues et al., 2016) or by using two separate encoders for src and mt and passing the concatenation of both encoders' final states to the decoder (Libovický et al., 2016). A small number of multi-source neural APE approaches were proposed in the WMT 2017 APE shared task. The two-encoder architecture (Junczys-Dowmunt and Grundkiewicz, 2017; Chatterjee et al., 2017; Varis and Bojar, 2017) of multi-source models utilizes both the source text (src) and the MT output (mt) to predict the post-edited output (pe) in a single end-to-end neural architecture.

In the WMT 2018 APE shared task, further multi-source APE architectures based on the transformer model (Vaswani et al., 2017) have been presented. The winning team for the NMT task in WMT 2018 Tebbifakhr et al. (2018) employ sequence-level loss functions in order to avoid exposure bias during training and to be consistent with the automatic evaluation metrics. (Pal et al., 2018) proposed an APE model that uses two separate self-attention-based encoders to encode mt and src, followed by a self-attended joint encoder that attends over a combination of the two encoded sequences and is used by the decoder for generating the post-edited sentence pe. Shin and Lee (2018) propose that each encoder has its own self-attention and feed-forward layer to process each input separately. On the decoder side, they add two additional multi-head attention layers, one for $src \rightarrow mt$ and another for $src \rightarrow pe$. There-

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 3: Shared Task Papers (Day 2) pages 124–131

after another multi-head attention between the output of those attention layers helps the decoder to capture common words in mt which should remain in pe. The WMT 2018 winner for the PB-SMT task (Junczys-Dowmunt and Grundkiewicz, 2018) also presented transformer-based multi-source APE called a dual-source transformer architecture. They use two encoders and stack an additional cross-attention component for $src \rightarrow pe$ above the previous cross-attention for $mt \rightarrow pe$. Comparing Shin and Lee (2018)'s approach with the winner system, there are only two differences in the architecture: (i) the cross-attention order of $src \rightarrow mt$ and $src \rightarrow pe$ in the decoder, and (ii) the winner system additionally shares parameters between two encoders.

In this work, we present a multi-source neural APE architecture called *transference*[1]. Our model contains (i) a source encoder (enc_{src}) which encodes src information, (ii) a second encoder ($enc_{src \rightarrow mt}$) which can also be viewed as a standard transformer decoding block, however, without masking, and (iii) a decoder (dec_{pe}) which captures the final representation from $enc_{src \rightarrow mt}$ via cross-attention. We thus recombine the different blocks of the transformer architecture and repurpose them for the APE task in a simple yet effective way. The intuition behind our architecture is to generate better representations via both self- and cross-attention and to further facilitate the learning capacity of the feed-forward layer in the decoder block.

The rest of the paper is organized as follows. In 2, we describe the *transference* architecture; 3 describes our experimental setup; 4 reports the results of our approach against the baseline; and finally, 5 concludes the paper with directions for future work.

2 Transference Model for APE

We propose a multi-source transformer model called *transference* (Figure 1), which takes advantage of both the encodings of src and mt and attends over a combination of both sequences while generating the post-edited sentence. The second encoder, $enc_{src \rightarrow mt}$, is identical to the transformer's decoder block but uses no masking in the self-attention layer, thus having one self-attention

[1]Our implementation is available at https://github.com/santanupal1980/Transference.git

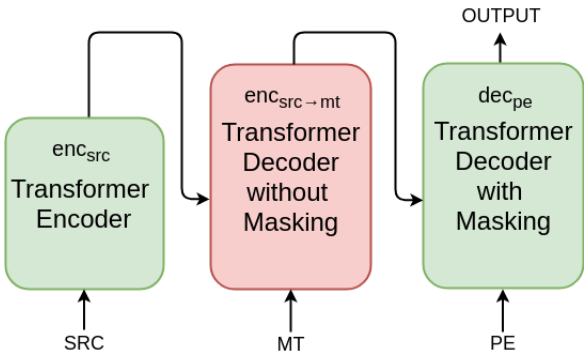

Figure 1: The *transference* model architecture for APE ($\{\mathbf{src}, \mathbf{mt}\}_{\mathbf{tr}} \rightarrow \mathbf{pe}$).

layer and an additional cross-attention layer for $src \rightarrow mt$. Here, the enc_{src} encoder and the dec_{pe} decoder are equivalent to the original transformer for neural MT (Vaswani et al., 2017). Put differently, our multi-source APE implementation extends Vaswani et al. (2017) by introducing an additional encoding block by which src and mt communicate with the decoder.

3 Experiments

We compare our approach against the *raw MT* output provided by the 1st-stage MT system. We evaluate the systems using BLEU (Papineni et al., 2002) and TER (Snover et al., 2006).

3.1 Data

For our experiments, we use the English–German WMT 2019 (Chatterjee et al., 2018) neural APE data. All released APE datasets consist of English–German triplets containing source English text (src) from the IT domain, the corresponding German translations (mt) from a 1st-stage NMT system, and the corresponding human-post-edited version (pe). Table 1 presents the statistics of the released data. As this released APE dataset is small in size (see Table 1), the synthetic eScape APE corpus (Negri et al., 2018), consisting of more than 7M triples, is available as an additional resource. All datasets, except for the eScape corpus, do not require any preprocessing in terms of encoding, tokenization or alignment.

For cleaning the noisy eScape dataset containing many unrelated language words (e.g. Chinese), we perform the following two steps: (i) we use the cleaning process described in Pal et al. (2015), and (ii) we execute the Moses (Koehn et al., 2007) corpus cleaning scripts with minimum and maximum number of tokens set to 1 and 100, respectively.

Corpus	Sentences	
	Overall	Cleaning
Train	13,442	-
Dev	1,000	-
Test	1,023	-
eScape	7.2M	6.5M

Table 1: Statistics of the WMT 2019 English-German APE Shared Task Dataset.

(iii) After cleaning, we perform punctuation normalization, and then use the Moses tokenizer to tokenize the eScape corpus with 'no-escape' option. Finally, we apply true-casing.

3.2 Experiment Setup

We split the released data (13.4K) into two sets; we use the first 12K for training and the remaining 1.4K as validation data. The development set (Dev) released by WMT2019[2] is used as test data for our experiment. We build two models *transference4M* and *transferenceALL* using slightly different training procedures.

For *transference4M*, we first train on a training set called eScape4M combined with the first 12k of the provided NMT training data. This eScape4M data is prepared using in-domain (for our case the 12K training data) bilingual cross-entropy difference for data selection as described in Axelrod et al. (2011). The difference in cross-entropy is computed based on two language models (LM): a domain-specific LM is estimated from the in-domain (12K) PE corpus (lm_i) and the out-domain LM (lm_o) is estimated from the eScape corpus. We rank the eScape corpus by assigning a score to each of the individual sentences which is the sum of the three cross-entropy (H) differences. For a j^{th} sentence pair src_j–mt_j–pe_j, the score is calculated based on Equation 1.

$$\begin{aligned} score = &|H_{src}(src_j, lm_i) - H_{src}(src_j, lm_o)| \\ &+ |H_{mt}(mt_j, lm_i) - H_{mt}(mt_j, lm_o)| \\ &+ |H_{pe}(pe_j, lm_i) - H_{pe}(pe_j, lm_o)| \quad (1) \end{aligned}$$

For *transferenceALL*, we initially train on the complete eScape dataset (eScapeAll) combined with the first 12k of the training data. The eScapeAll data is sorted based on their in-domain similarities as described in Equation 1.

[2]It is to be noted that, the released development set and test set are same as in WMT2018.

Both models are then fine-tuned towards the real data, by training again solely on the first 12k segments of the provided data. For both models, we perform checkpoint averaging using the 8 best checkpoints. We report the results on the development set provided by WMT2019, which we use as a test set.

To handle out-of-vocabulary words and to reduce the vocabulary size, instead of considering words, we consider subword units (Sennrich et al., 2016) by using byte-pair encoding (BPE). In the preprocessing step, instead of learning an explicit mapping between BPEs in the src, mt and pe, we define BPE tokens by jointly processing all triplets. Thus, src, mt and pe derive a single BPE vocabulary. Since mt and pe belong to the same language (DE) and src is a close language (EN), they naturally share a good fraction of BPE tokens, which reduces the vocabulary size.

3.3 Hyper-parameter Setup

We follow a similar hyper-parameter setup for all reported systems. All encoders (for $\{src, mt\}_{tr} \rightarrow pe$), and the decoder, are composed of a stack of $N_{src} = N_{mt} = N_{pe} = 6$ identical layers followed by layer normalization. We set all dropout values in the network to 0.1. During training, we employ label smoothing with value ϵ_{ls} = 0.1. The learning rate is varied throughout the training process, and increasing for the first training steps $warmup_{steps} = 8000$ and afterwards decreasing as described in (Vaswani et al., 2017). All remaining hyper-parameters are set analogously to those of the transformer's *base* model.

At training time, the batch size is set to 25K tokens, with a maximum sentence length of 256 subwords, and a vocabulary size of 28K. After each epoch, the training data is shuffled. During decoding, we perform beam search with a beam size of 4. We use shared embeddings between mt and pe in all our experiments.

4 Results

The results of our two models, *transference4M* and *transferenceALL*, in comparison to the baseline *raw MT* are presented in Table 2 and 3. Table 2 reports results on the WMT2019 development set (Dev), Table 3 on the WMT2019 test set (Test).

Exp No.	Models	Dev	
		BLEU ↑	TER ↓
Baseline			
1	*raw MT*	76.76	15.08
No fine-tuning			
2	*transference4M* (CONTRASTIVE)	77.11 (+0.35)	14.94 (-0.14)
3	*transferenceALL*	77.25 (+0.49)	14.87 (-0.21)
Fine tune with 12K			
4	*transference4M*	77.22 (+0.46)	14.89 (-0.19)
5	*transferenceALL*	77.39 (+0.63)	14.71 (-0.37)
Average 8 checkpoints on fine tuned models			
6	*transference4M*	77.27 (+0.51)	14.88 (-0.20)
7	*transferenceALL* (PRIMARY)	**77.67 (+0.91)**	**14.52 (-0.56)**

Table 2: Evaluation results on the WMT APE 2019 development set for the EN-DE NMT task.

Exp No.	Models	Test	
		BLEU ↑	TER ↓
Baseline			
1	*raw MT*	74.73	16.84
Submission			
2	*transference4M* (CONTRASTIVE)	73.97 (-0.76)	17.31 (+0.47)
3	*transferenceALL* (PRIMARY)	**75.75 (+1.02)**	**16.15 (-0.69)**

Table 3: Evaluation results on the WMT APE 2019 test set for the EN-DE NMT task.

4.1 Baselines

The *raw MT* output in Table 2 and Table 3 is a strong black-box NMT system (i.e., 1st-stage MT) on Dev and Test respectively. We report its performance observed with respect to the ground truth (pe), i.e., the post-edited version of mt. The original MT system scores 76.76 BLEU points and 15.08 TER on Dev as well as 74.73 BLEU points and 16.84 TER on Test.

4.2 Transference Transformer for APE

Table 2 shows the results of our *transference* architecture on the Dev set, where our two experimental setups *transference4M* (Exp 2) and *transferenceALL* (Exp 3) improve the performance over the baseline system. Compared to *transference4M* (Exp 2), our *transferenceALL* (Exp 3) performs better in terms of both BLEU and TER on the Dev set. Moreover, fine-tuning our transference models (Exp 4 and 5 in Table 2) yields further performance gains. Additionally averaging the 8 best checkpoints of our fine-tuned version models (Exp 6 and 7) provides further improvements. All models except *transference4M* (CONTRASTIVE, our *contrastive* submission in WMT2019 APE task) yield statistically significant results ($p < 0.001$) over the *raw MT* baseline. *transferenceALL* (PRIMARY, our *primary* submission in WMT2019 APE task) (Exp 7) also provides statistically significant improvement over *transference4M* (Exp 6). For these and all following significance tests we employ the method by Clark et al. (2011)[3]. Table 2 shows that our APE architecture *transferenceALL* (PRIMARY) (Exp 7) significantly improves over the already very good NMT system by about +0.91 BLEU and -0.56 TER.

Table 3 presents the results of our submissions on the Test set in the WMT 2019 EN-DE APE task. We submitted *transference4M* (CONTRASTIVE) system – a weak model having performance close to the baseline, (i) to check whether in-domain data provides any gain in performance on the Test set or not, (ii) to create another baseline trained on in-domain data, by which we could analyze our PRIMARY transference model's capability of transfer learning. So far, we could not find an explanation why our CONTRASTIVE system behaves completely different on the Test set compared to the Dev

[3] https://github.com/jhclark/multeval

set. However, our primary submission *transferenceALL* (PRIMARY) shows similar performance on the WMT2019 Test set as on the Dev set. Overall our *transferenceALL* (PRIMARY) submission achieves statistically significant +1.02 absolute BLEU point and -0.69 absolute in TER improvements in performance over the baseline on the Test set.

4.3 Discussion

It is important to note that raw MT provides a strong baseline. Our proposed *transference* model (*transferenceALL*) shows statistically significant improvements in terms of BLEU and TER compared to this baseline even before fine-tuning, and further improvements after fine-tuning. Finally, after averaging the 8 best checkpoints, our *transferenceALL* model also shows consistent improvements in comparison to the baseline and other experimental setups.

Table 4 shows the performance of our *transferenceALL* model compared to the winner system of WMT 2018 ($wmt18_{Best}$) for the NMT task (Tebbifakhr et al., 2018) on Dev and Test data. The primary submission of $wmt18_{Best}$ scores 14.78 in TER and 77.74 in BLEU on the Dev set and 16.46 in TER and 75.53 in BLEU on the Test set. In comparison to $wmt18_{Best}$, our *transferenceALL* model achieves better scores in TER on both the Dev and Test set, however, in terms of BLEU the score acquired by our *transferenceALL* model is slightly worse for the Dev set, while some improvements were achieved on the Test data. In comparison to the $wmt2019_{Best}$ system, which achieved 16.06 in TER and 75.95 in BLEU according to the official released results[4], we do not used BERT (Devlin et al., 2018) in our system. Even though $wmt2019_{Best}$ integrated BERT, there is no statistical significant performance difference to our primary submission. Moreover, our system does not perform ensembling of multiple models, as the 2^{nd} best system in WMT 2019, which achieves 16.11 in TER and 76.22 in BLEU.

We believe the reasons for the effectiveness of our approach to be as follows. (1) Our $enc_{src \rightarrow mt}$ contains two attention mechanisms: one is self-attention and another is cross-attention. The self-attention layer is not masked here; therefore, the cross-attention layer in $enc_{src \rightarrow mt}$ is informed by both previous and future time-steps from the self-attended representation of mt (enc_{mt}) and additionally from enc_{src}. As a result, each state representation of $enc_{src \rightarrow mt}$ is learned from the context of src and mt. This might produce better representations for dec_{pe} which can access the combined context. In contrast, in $wmt18_{Best}$, the dec_{pe} accesses the concatenated encoded representations from src and mt encoder jointly. (2) Since pe is a post-edited version of mt, sharing the same language, mt and pe are quite similar compared to src. Therefore, attending over a fine-tuned representation from mt along with src, which is what we have done in this work, might be a reason for the better results compared to those achieved by attending over concatenated encoded information from src and mt directly.

[4] http://www.statmt.org/wmt19/ape-task.html

5 Conclusions and Future Work

In this paper, we presented our submissions to the APE shared task at WMT 2019. We extend the transformer-based architecture to a multi-encoder transformer-based model that extends the standard transformer blocks in a simple and effective way for the APE task. Our model makes use of two separate encoders to encode src and mt; the second encoder additionally attends over a combination of both sequences to prepare the representation for the decoder to generate the post-edited translation. The proposed model outperforms the best-performing system of WMT 2018 on the Test data. Our primary submission ranked 3^{rd}, however compared to other two top systems, the performance differences are not statistically significant.

Taking a departure from traditional transformer-based encoders, which perform self-attention only, our second encoder also performs cross-attention to produce representations for the decoder based on both src and mt. Our proposed multi-encoder transformer-based architecture is also generic and can be used for any multi-modal (or multi-source) task, e.g., multi-modal translation, multi-modal summarization.

Acknowledgments

This research was funded in part by the German research foundation (DFG) under grant number GE 2819/2-1 (project MMPE) and the German Federal Ministry of Education and Research (BMBF) under funding code 01IW17001 (project Deeplee). The responsibility for this publication

Models	Dev		Test	
	BLEU ↑	TER ↓	BLEU ↑	TER ↓
$wmt2018_{Best}$	77.74	14.78	75.53	16.46
transferenceALL	77.67 (-0.07)	**14.52 (-0.26)**	**75.75 (+0.22)**	**16.15 (-0.31)**

Table 4: Comparison with $wmt2018_{Best}$ on the WMT APE 2018 Dev/Test set for the EN-DE NMT task.

lies with the authors. We also want to thank the reviewers for their valuable input, and the organizers of the shared task. We also thank the NVIDIA Corporation for providing a GPU through the NVIDIA GPU Grant.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain Adaptation via Pseudo In-domain Data Selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Rajen Chatterjee, M. Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-Source Neural Automatic Post-Editing: FBK's participation in the WMT 2017 APE shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 630–638, 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*, Brussels, Belgium. Association for Computational Linguistics.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 176–181, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. The AMU-UEdin Submission to the WMT 2017 Shared Task on Automatic Post-Editing. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 639–646, Copenhagen, Denmark. 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.

Kevin Knight and Ishwar Chander. 1994. Automated Postediting of Documents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 1)*, AAAI '94, pages 779–784, Seattle, Washington, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 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, Prague, Czech Republic.

Jindřich Libovický, Jindřich Helcl, Marek Tlustý, Ondřej Bojar, and Pavel Pecina. 2016. CUNI System for WMT16 Automatic Post-Editing and Multimodal Translation Tasks. In *Proceedings of the First Conference on Machine Translation*, pages 646–654, Berlin, Germany. Association for Computational Linguistics.

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).

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-Translation for Neural Machine Translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1828–1836, Osaka, Japan. The COLING 2016 Organizing Committee.

Santanu Pal, Nico Herbig, Antonio Krger, and Josef van Genabith. 2018. A Transformer-Based Multi-Source Automatic Post-Editing System. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 840–848, Belgium, Brussels. Association for Computational Linguistics.

Santanu Pal, Sudip Naskar, and Josef van Genabith. 2015. UdS-sant: English–German hybrid machine translation system. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 152–157, Lisbon, Portugal. Association for Computational Linguistics.

Santanu Pal, Sudip Kumar Naskar, and Josef van Genabith. 2016a. Multi-Engine and Multi-Alignment Based Automatic Post-Editing and Its Impact on Translation Productivity. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2559–2570, Osaka, Japan.

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, and Josef van Genabith. 2016b. 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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, Pennsylvania.

Carla Parra Escartín and Manuel Arcedillo. 2015a. Living on the Edge: Productivity Gain Thresholds in Machine Translation Evaluation Metrics. In *Proceedings of the Fourth Workshop on Post-editing Technology and Practice*, pages 46–56, Miami, Florida (USA). Association for Machine Translation in the Americas (AMTA).

Carla Parra Escartín and Manuel Arcedillo. 2015b. Machine Translation Evaluation Made Fuzzier: A Study on Post-Editing Productivity and Evaluation Metrics in Commercial Settings. In *Proceedings of the MT Summit XV*, Miami (Florida). International Association for Machine Translation (IAMT).

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Jaehun Shin and Jong-Hyeok Lee. 2018. Multi-encoder Transformer Network for Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 853–858, Belgium, Brussels. Association for Computational Linguistics.

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 Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

Amirhossein Tebbifakhr, Ruchit Agrawal, Rajen Chatterjee, Matteo Negri, and Marco Turchi. 2018. Multi-Source Transformer with Combined Losses for Automatic Post Editing. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 859–865, Belgium, Brussels. Association for Computational Linguistics.

Dusan Varis and Ondřej Bojar. 2017. CUNI System for WMT17 Automatic Post-Editing Task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 661–666, Copenhagen, Denmark. Association for Computational Linguistics.

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 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.