A Transformer-Based Multi-Source Automatic Post-Editing System

Santanu Pal[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
{nico.herbig, krueger, josef.van_genabith}@dfki.de

Abstract

This paper presents our English–German Automatic Post-Editing (APE) system submitted to the APE Task organized at WMT 2018 (Chatterjee et al., 2018). The proposed model is an extension of the transformer architecture: two separate self-attention-based encoders encode the machine translation output (mt) and the source (src), followed by a joint encoder that attends over a combination of these two encoded sequences (enc_{src} and enc_{mt}) for generating the post-edited sentence. We compare this multi-source architecture (i.e, $\{src, mt\} \rightarrow pe$) to a monolingual transformer (i.e., $mt \rightarrow pe$) model and an ensemble combining the multi-source $\{src, mt\} \rightarrow pe$ and single-source $mt \rightarrow pe$ models. For both the PBSMT and the NMT task, the ensemble yields the best results, followed by the multi-source model and last the single-source approach. Our best model, the ensemble, achieves a BLEU score of 66.16 and 74.22 for the PBSMT and NMT task, respectively.

1 Introduction & Related Work

The ultimate goal of machine translation (MT) is to provide fully automatic publishable quality translations. However, state-of-the-art MT systems often fail to deliver this; translations produced by MT systems contain different errors and require human interventions to post-edit the translations. Nevertheless, MT has become a standard in the translation industry as post-editing on MT output is often faster and cheaper than performing human translation from scratch.

APE is a method that aims to automatically correct errors made by MT systems before performing actual human-post-editing (PE) (Knight and Chander, 1994), thereby reducing the translators' workload and increasing productivity (Parra Escartín and Arcedillo, 2015b,a; Pal et al., 2016a). Various automatic and semi-automatic techniques have been developed to auto-correct repetitive errors (Roturier, 2009; TAUS/CNGL Report, 2010). The advantage of APE lies in its capability to adapt to any black-box (first-stage) MT engine; i.e., upon availability of human-corrected post-edited data, no incremental training or full re-training of the first-stage MT system is required to improve the overall translation quality. APE can therefore 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 may additionally make use of the source (src). A more detailed motivation on APE can be found in 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. In general, the field of APE covers a wide methodological range, including SMT-based approaches (Simard et al., 2007a,b; Lagarda et al., 2009; Rosa et al., 2012; Pal et al., 2016c; Chatterjee et al., 2017b), and neural APE (Pal et al., 2016b; Junczys-Dowmunt and Grundkiewicz, 2016; Pal et al., 2017) based on neural machine translation (NMT). Some of the state-of-the-art multi-source approaches, both statistical (Béchara et al., 2011; Chatterjee et al., 2015) and recently neural (Libovický et al., 2016; Chatterjee et al., 2017a; Junczys-Dowmunt and Grundkiewicz, 2016; Varis and Bojar, 2017), explore learning from $\{src, mt\} \rightarrow pe$ (multi-source, MS) to take advantage of the dependencies of translation errors in mt originating from src.

Exploiting source information in multi-source neural APE 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 few approaches to multi-source neural APE have been proposed in the WMT-2017 APE shared task. Junczys-Dowmunt and Grundkiewicz (2017) explore different combinations of attention mechanisms including soft attention and hard monotonic attention on an end-to-end neural APE model that combines both mt and src in a single neural architecture. Chatterjee et al. (2017a) extend the two-encoder architecture of multi-source models presented in Libovický et al. (2016). In their extension each encoder concatenates both contexts having their own attention layer that is used to compute the weighted context of src and mt. Finally, a linear transformation is applied on the concatenation of both weighted contexts. Varis and Bojar (2017) implement and compare two multi-source architectures: In the first setup, they use a single encoder with concatenation of src and mt sentences, and in the second setup, they use two character-level encoders for mt and src, separately, along with a character-level decoder. The initial state of this decoder is a weighted combination of the final states of the two encoders.

Intuitively, such an integration of source-language information in APE should be useful in conveying the context information to improve the APE performance. To provide the awareness of errors in mt originating from src, the transformer architecture (Vaswani et al., 2017), which is built solely upon attention mechanisms (Bahdanau et al., 2015), makes it possible to model dependencies without regard to their distance in the input or output sequences and also captures global dependencies between input and output (for our case src, mt, and pe). The transformer architecture replaces recurrence and convolutions by using positional encodings on both the input and output sequences. The encoder and decoder both use multi-head (facilitating parallel computations) self-attention to compute representations of their corresponding inputs, and also compute multi-head vanilla-attentions between encoder and decoder representations.

Our APE system extends this transformer-based NMT architecture (Vaswani et al., 2017) by using two encoders, a joint encoder, and a single decoder. Our model concatenates two separate self-attention-based encoders (enc_{src} and enc_{mt}) and passes this sequence through another self-attended joint encoder ($enc_{src,mt}$) to ensure capturing dependencies between src and mt. Finally, this joint encoder is fed to the decoder which follows a similar architecture as described in Vaswani et al. (2017). The entire model is optimized as a single end-to-end transformer network.

2 Transformer-Based Multi-Source APE

MT errors originating from the input source sentences suggest that APE systems should leverage information from both the src and mt, instead of considering mt in isolation. This can help the model to disambiguate corrections applied at every time step. Generating the pe output from mt is greatly facilitated by the availability of src. To achieve benefits from both ***single-source*** ($\mathbf{mt} \rightarrow \mathbf{pe}$) and ***multi-source*** ($\{\mathbf{src}, \mathbf{mt}\} \rightarrow \mathbf{pe}$) APEs, our primary submission in the WMT 2018 shared task is an ensemble of these two models.

Transformer-based models are currently providing state-of-the-art performance in MT; hence, we want to explore a similar architecture for this year's APE task. We extend the transformer architecture to investigate how efficient this approach is in a multi-source scenario. In a MT task, it was already shown that a transformer can learn long-range dependencies. Therefore, we explore if we can leverage information from src and mt via a joint encoder through self-attention (see Section 2.2) to provide dependencies between src–mt that are then projected to the pe.

To investigate this, we implement and evaluate three different models: a single-source approach, a multi-source approach, and an ensemble of both, described in more detail below.

2.1 Single-Source Transformer for APE ($\mathbf{mt} \rightarrow \mathbf{pe}$)

Our single-source model (SS) is based on an encoder-decoder-based transformer architecture (Vaswani et al., 2017). Transformer models can replace sequence-aligned recurrence entirely and follow three types of multi-head attention: encoder-decoder attention (also known as vanilla

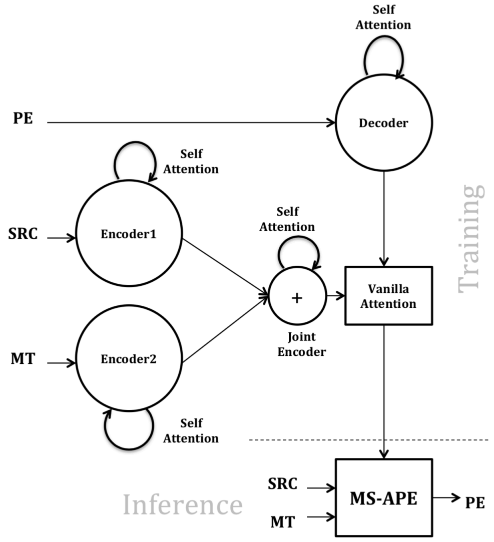

Figure 1: Multi-source transformer-based APE

attention), encoder self-attention, and masked decoder self-attention. Since for multi-head attention each head uses different linear transformations, it can learn these separate relationships in parallel, thereby improving learning time.

2.2 Multi-source Transformer for APE ($\{\mathbf{src}, \mathbf{mt}\} \rightarrow \mathbf{pe}$)

For our multi-source model (MS), we propose a novel joint transformer model (cf. Figure 1), which combines the encodings of src and mt and attends over a combination of both sequences while generating the post-edited sentence. Apart from enc_{src} and enc_{mt}, each of which is equivalent to the original transformer's encoder (Vaswani et al., 2017), we use a joint encoder with an equivalent architecture, to maintain the homogeneity of the transformer model. For this, we extend Vaswani et al. (2017) by introducing an additional identical encoding block by which both the enc_{src} and the enc_{mt} encoders communicate with the decoder.

Our multi-source neural APE computes intermediate states $\mathbf{enc_{src}}$ and $\mathbf{enc_{mt}}$ for the two encoders, $\mathbf{enc_{src,mt}}$ for their combination, and $\mathbf{dec_{pe}}$ for the decoder in sequence-to-sequence modeling. One self-attended encoder for src maps $\mathbf{s} = (s_1, s_2, ..., s_k)$ into a sequence of continuous representations, $\mathbf{enc_{src}} = (e_1, e_2, ..., e_k)$, and a second encoder for mt, $\mathbf{m} = (m_1, m_2, ..., m_l)$, returns another sequence of continuous representations, $\mathbf{enc_{mt}} = (e_1^{'}, e_2^{'}, ..., e_l^{'})$. The self-attended joint encoder (cf. Figure 1) then receives the concatenation of $\mathbf{enc_{src}}$ and $\mathbf{enc_{mt}}$, $\mathbf{enc_{concat}} = [\mathbf{enc_{src}}, \mathbf{enc_{mt}}]$ as an input, and passes it through the stack of 6 layers, with residual connections, a self-attention and a position-wise fully connected feed-forward neural network. As a result, the joint encoder produces a final representation ($\mathbf{enc_{src,mt}}$) for both src and mt. Self-attention at this point provides the advantage of aggregating information from all of the words, including src and mt, and successively generates a new representation per word informed by the entire src and mt context. The decoder generates the pe output in sequence, $\mathbf{dec_{pe}} = (p_1, p_2, \ldots, p_n)$, one word at a time from left to right by attending previously generated words as well as the final representations ($\mathbf{enc_{src,mt}}$) generated by the encoder.

2.3 Ensemble

In order to leverage the network architecture for both single-source and multi-source APE as discussed above, we decided to **ensemble** several expert neural models. Each model is averaged using the 5 best saved checkpoints, which generate different translation outputs. Taking into account all these generated translation outputs, we implement an ensemble technique based on the frequency of occurrence of the output words. Corresponding to each input word, we calculate the most frequent occurrence of the output word from all the generated translation outputs. For the two different APE tasks, we ensemble the following models:

- PBSMT task: We ensemble a SS ($mt \rightarrow pe$) and a MS ($\{src, mt\} \rightarrow pe$) average model.
- NMT task: We ensemble two average SS ($mt \rightarrow pe$) and MS ($\{src, mt\} \rightarrow pe$) models, together with a SS and a MS model that are fine-tuned on a subset of the training set (cf. Section 3.3.2).

3 Experiments

In our experiment we investigate (1) how well the transformer-based APE architecture performs in general, (2) if our multi-source architecture using the additional joint encoder improves the performance over a single-source architecture, and (3) if ensembling of single-source and multi-source architectures facilitates APE even further.

3.1 Data

Since this year's WMT 2018 APE task (Chatterjee et al., 2018) is divided into two sub-tasks, different datasets are provided for each task: for the PBSMT task, there is a total of 23K English–German APE data samples (11K from WMT 2016 and 12K from WMT 2017) (Bojar et al., 2017). For the NMT task, 13,442 samples of English–German APE data are provided.

All released APE data consists of English–German triplets containing source English text (src) from the IT domain, the corresponding German translations (mt) from a first stage MT system, and the corresponding human-post-edited version (pe), all of them already tokenized. As this released APE dataset is small in size (see Table 1), additional resources are also available: first, the 'artificial training data' (Junczys-Dowmunt and Grundkiewicz, 2016) containing 4.5M sentences, 4M of which are weakly similar to the WMT 2016 training data, while 500K show very similar TER statistics; and second, the synthetic 'eSCAPE' APE corpus (Negri et al., 2018), consisting of more than 7M triples for both NMT and PBSMT.

Table 1 presents the statistics of the released data for the English–German APE Task organized in WMT 2018. These datasets, except for the eSCAPE corpus, do not require any preprocessing in terms of encoding 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 80, respectively. After cleaning, we use the Moses tokenizer to tokenize the eSCAPE corpus. To handle out-of-vocabulary words, words are preprocessed into subword units (Sennrich et al., 2016) using byte-pair encoding (BPE).

3.2 Hyper-Parameter Settings

For $\{\mathbf{src}, \mathbf{mt}\} \rightarrow \mathbf{pe}$, both the self-attended encoders, the joint encoder, and the decoder are composed of a stack of $N = 6$ identical layers followed by layer normalization. Each layer again consists of two sub-layers and a residual connection (He et al., 2016) around each of the two sub-layers. During training, we employ label smoothing of value ϵ_{ls} = 0.1. The output dimension produced by all sub-layers and embedding layers is defined as $d_{model} = 256$. All dropout values in the network are set to 0.2. Each encoder and decoder contains a fully connected feed-forward network having dimensionality $d_{model} = 256$ for the input and output and dimensionality $d_{ff} = 1024$ for the inner layer. This is a similar setting to Vaswani et al. (2017)'s $C - model$[1]. For the scaled dot-product attention, the input consists of queries and keys of dimension d_k, and values of dimension d_v. As multi-head attention parameters, we employ $h = 8$ for parallel attention layers, or heads. For each of these we use a dimensionality of $d_k = d_v = d_{model}/h = 32$. For optimization, we use the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. The learning rate is varied throughout the training process, first increasing linearly for the first training steps $warmup_{steps} = 4000$ and then adjusted as described in (Vaswani et al., 2017).

At training time, the batch size is set to 32 samples, with a maximum sentence length of 80 subwords, and a vocabulary of the 50K most frequent subwords. After each epoch, the training data is shuffled. For encoding the word order, our model uses learned positional embeddings (Gehring et al., 2017), since Vaswani et al. (2017) reported nearly identical results to sinusoidal encodings. After finishing training, we save the 5 best checkpoints saved at each epoch. Finally, we use a single model obtained by averaging the last 5 checkpoints. During decoding, we perform greedy-search-based decoding.

We follow a similar hyper-parameter setup for $mt \rightarrow pe$. The total number of parameters for our $\{src, mt\} \rightarrow pe$ and $mt \rightarrow pe$ model is 46×10^6 and 28×10^6, respectively.

3.3 Experiment Setup

In this section, we present the training process, using the above datasets, to train $mt \rightarrow pe$, $\{src, mt\} \rightarrow pe$, and ensemble models for both PBSMT and NMT.

3.3.1 PBSMT Task

For PBSMT, we first train both our SS and MS systems with the cleaned eSCAPE corpus for 3 epochs. We then perform transfer learning with 4M artificial data for 7 epochs. Afterwards, fine-tuning is performed using the 500K artificial and 23K real PE training data for another 20 epochs.

[1] Note: at the time of submission we couldn't test the *Transformer (big)* model due to unavailability of enough computation power

	Sentences				
	Corpus	2016	2017	2018	Cleaning
PBSMT	Train	12,000	11,000	-	-
	Dev	1,000	-	-	-
	Test	2,000	2,000	2,000	-
NMT	Train	-	-	13,442	-
	Dev	-	-	1,000	-
	Test	-	-	1,023	-
Additional Resources	Artificial	-	4M + 500K	-	-
	eSCAPE-PBSMT	-	-	7,258,533	6,521,736
	eSCAPE-NMT	-	-	7,258,533	6,485,507

Table 1: Statistics of the WMT 2018 APE Shared Task Dataset.

Furthermore, we generate an ensemble model, by averaging the 5 best checkpoints of SS with the 5 best checkpoints of MS.

We use the WMT 2016 development data (dev2016) containing 1,000 triplets to validate the model during training. To test our system performance, we use the WMT 2016 and 2017 test data (test2016, test2017), each containing 2,000 triplets. Furthermore, we report the results of the submitted ensemble model on test2018.

3.3.2 NMT Task

Initial tests for pre-training our NMT model on the NMT eSCAPE data showed no performance improvements. Therefore, we use the PBSMT SS and MS models as a basis for the NMT task. We use the PBSMT models after training them on the eSCAPE corpus, the 4M artificial data and the 500K + 23K train sets of WMT 16 and 17. These SMT-based models are then fine-tuned using the WMT 2018 NMT APE data (train18) for 60 epochs.

Afterwards, we perform an additional fine-tuning step towards the dev18/test18 dataset: For this, we extract sentences of train18 that are similar to the sentences contained in dev18/test18 and fine-train for another 15 epochs on this subset of train18, which we call fine-tune18. As a similarity measure we use the cosine similarity between the train src and mt segments and the test src and mt segments, respectively. These cosine similarities for src and mt are then simply multiplied to achieve an overall similarity measure. Our fine-tuning dataset contains only sentences with an overall similarity of at least 0.9.

Last, two separate ensemble models are created. One consists of only the non-fine-tuned SS and MS models, and one ensembles the SS and MS models in both fine-tuned and non-fine-tuned variants. Both ensembles are created by averaging over the 5 best checkpoints of each sub-model.

We report the results of all created models for the dev18 NMT dataset, and additionally those of the submitted overall ensemble model on test18.

3.4 Results and Discussion

Table 2 presents the results for the PBSMT APE task (cf. 3.3.1), where two different transformer-based models, one ensemble of these models and the baseline BLEU scores are shown. The baseline here refers to the original MT output evaluated with respect to the corresponding PE translation. All models yield statistically significant results ($p < 0.001$) over this baseline. MS_{avg} also provides statistically significant improvement over SS_{avg}. For this and all following significance tests we employ the method by Clark et al. (2011)[2].

Generally, reasons for the good performance of our transformer-based MS architecture in comparison to the SS approach for PBSMT-based APE could be the positional encoding that injects information about the relative or absolute position of the tokens in the sequence. This might help to handle word order errors in mt for a given src context. Another possible explanation lies in the self-attention mechanism, which handles local word dependencies for src, mt, and pe. After the individual dependencies are learned by the two encoders' self-attention mechanisms, another level of self-attention is performed that can jointly learn from both src and mt using our joint encoder, thereby informing the decoder about the long-range dependencies between the words within both src and mt. Compared to RNNs, we believe that this technique can better convey source information via mt to the decoder. The ensemble model then leverages the advantages of both our SS and MS approaches to further improve the results.

The results for our transformer-based architec-

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

WMT APE Systems	eScape	4M	500K	train16	train17	test16	test17	test18
Baseline	-					62.92	62.11	62.99
MS_{avg}	3 eps	7 eps	20 eps			67.31	67.66	-
SS_{avg}	3 eps	7 eps	20 eps			66.27	66.60	-
Ensemble	$MS_{avg\{5cps\}}$ + $SS_{avg\{5cps\}}$					68.52	68.91	66.16

Table 2: Evaluation result of WMT 2018 PBSMT task for all trained models.

WMT APE Systems	Base Model	train18	fine-tune18	dev18	test18
Baseline	-	-	-	76.66	74.73
MS_{avg}	MS_{avg} (PBSMT)	60 eps	-	74.84	-
SS_{avg}	SS_{avg} (PBSMT)	60 eps	-	72.75	
$MS_{finetuned}$	MS_{avg} (NMT)	-	15 eps	75.05	-
$SS_{finetuned}$	SS_{avg} (NMT)	-	15 eps	73.17	-
$Ensemble$	$MS_{avg\{5cps\}}$ + $SS_{avg\{5cps\}}$			75.80	-
$Ensemble_{finetuned}$	$MS_{avg\{5cps\}}$ + $SS_{avg\{5cps\}}$ + $MS_{finetuned\{5cps\}}$ + $SS_{finetuned\{5cps\}}$			75.96	74.22

Table 3: Evaluation result of WMT 2018 NMT task for all trained models.

ture for the NMT task are shown in Table 3. As can be seen, the baseline NMT system performs best, followed by the ensemble models, then the multi-source architectures and lastly the single-source approach. These differences between the three approaches, ensemble, MS, and SS, are all statistically significant. Fine-tuning provides some small, albeit insignificant, improvements over the non-fine-tuned versions.

While none of our architectures perform better than the baseline MT system for the NMT task, we clearly see that the multi-source approach helps, and that ensembling of different SS and MS models further increases the performance. These results are in line with our expectations, because intuitively, inspecting both src and mt should help detect and correct common errors. However, we are unsure why all models did not improve over the baseline, which could have been achieved by simply copying the mt. One reason might be the small amount of PE data, which comprises only 13K samples; this could also explain why the simple fine-tuning approach already leads to slightly higher BLEU scores. However, further human evaluation is necessary to better understand what our model is doing for the neural APE task and why it remains approximately 0.5 BLEU points below the baseline.

4 Conclusions and Future Work

In this paper, we investigated a novel transformer-based multi-source APE approach that jointly attends over a combination of src and mt to capture dependencies between the two. This architecture yields statistically significant improvements over single-source transformer-based models. An ensemble of both variants increases the performance further. For the PBSMT task, the baseline MT system was outperformed by 3.2 BLEU points, while the NMT baseline remains 0.51 BLEU points better than our APE approach on the 2018 test set.

In the future, we will investigate if the performance of each system can be improved by using a different hyper-parameter setup. Unfortunately, we could not test either the 'big' or the 'base' hyper-parameter configuration in Vaswani et al. (2017) due to unavailable computing resources at the time of submission. As additional future work, we would like to explore whether using re-ranking and ensembling of different neural APEs helps to improve the performance further. Moreover, we will incorporate word-level quality estimation features of mt into the encoding layer. Lastly, we will evaluate if our model indeed is able to better handle word order errors and to capture long-range dependencies, as we expect. Furthermore, we will analyze if adapting the learning rate to the size of the datasets used during training increases the performance compared to the currently used fixed learning rate initialization of 0.001.

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 lies with the authors. We also want to thank the reviewers for their valuable input, and the organizers of the shared task.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Hanna Béchara, Yanjun Ma, and Josef van Genabith. 2011. Statistical Post-Editing for a Statistical MT System. In *Proceedings of MT Summit XIII*, pages 308–315.

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. 2017a. 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, Gebremedhen Gebremelak, Matteo Negri, and Marco Turchi. 2017b. Online Automatic Post-editing for MT in a Multi-Domain Translation Environment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 525–535, Valencia, Spain. 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.

Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the Planet of the APEs: A Comparative Study of State-of-the-art Methods for MT Automatic Post-Editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 156–161, Beijing, China.

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.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. *CoRR*, abs/1705.03122.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

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.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. *ICLR*.

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: Interactive Poster and Demonstration Sessions*, pages 177–180, Prague, Czech Republic.

Antonio Lagarda, Vicent Alabau, Francisco Casacuberta, Roberto Silva, and Enrique Díaz-de Liaño. 2009. Statistical Post-editing of a Rule-based Machine Translation System. In *Proceedings of Human Language Technologies*, pages 217–220, Stroudsburg, PA, USA.

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, 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.

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, Qun Liu, and Josef van Genabith. 2017. Neural Automatic Post-Editing Using Prior Alignment and Reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 349–355, Valencia, Spain. Association for Computational Linguistics.

Santanu Pal, Marcos Zampieri, and Josef van Genabith. 2016c. USAAR: An Operation Sequential Model for Automatic Statistical Post-Editing. In *Proceedings of the First Conference on Machine Translation*, pages 759–763, Berlin, Germany.

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).

Rudolf Rosa, David Mareček, and Ondřej Dušek. 2012. DEPFIX: A System for Automatic Correction of Czech MT Outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 362–368, Stroudsburg, PA, USA.

Johann Roturier. 2009. Deploying Novel MT Technology to Raise the Bar for Quality: A Review of Key Advantages and Challenges. In *Proceedings of the twelfth Machine Translation Summit*.

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*.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007a. Statistical Phrase-based Post-Editing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 508–515, Rochester, New York.

Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. 2007b. Rule-based Translation With Statistical Phrase-based Post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206.

TAUS/CNGL Report. 2010. Machine Translation Post-Editing Guidelines Published. Technical report, TAUS.

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.