The Transference Architecture for Automatic Post-Editing

Santanu Pal[1], Hongfei Xu[1,2], Nico Herbig[2], Sudip Naskar[3],
Antonio Krüger[2], Josef van Genabith[1,2]
[1]Saarland University, Germany
[2]German Research Center for Artificial Intelligence (DFKI),
Saarland Informatics Campus, Germany
[3]Jadavpur University, Kolkata, India
{firstname.lastname}@uni-saarland.de
{firstname.lastname}@dfki.de

Abstract

In automatic post-editing (APE) it makes sense to condition post-editing (pe) decisions on both the source (src) and the machine translated text (mt) as input. This has led to multi-encoder based neural APE approaches. A research challenge now is the search for architectures that best support the capture, preparation and provision of src and mt information and its integration with pe decisions. In this paper we present an efficient multi-encoder based APE model, called *transference*. Unlike previous approaches, it (i) uses a transformer encoder block for src, (ii) followed by a 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. Our model outperforms the best performing systems by 1 BLEU point on the WMT 2016, 2017, and 2018 English–German APE shared tasks (PBSMT and NMT). Furthermore, the results of our model on the WMT 2019 APE task using NMT data shows performance at the level of the state-of-the-art. The inference time of our model is similar to the vanilla transformer-based NMT system although our model deals with two separate encoders. We further investigate the importance of our newly introduced second encoder and find that decreasing the number of layers hurts performance, while reducing the number of layers of the decoder does not matter much.

1 Introduction

Although machine translation (MT) systems are improving rapidly, the resulting translations may still require manual post-editing (PE) to achieve human-acceptable translation. Automatic post-editing (APE) is a method that aims to automatically correct errors in machine translated text before performing actual human post-editing (PE) (Knight and Chander, 1994), thereby reducing the post-editors' workload and increasing productivity (Pal et al., 2016a). APE systems learned from human PE data serve as downstream MT post-processing modules to improve the overall performance. 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 additionally using the source (src) has been shown to provide further benefits (Bojar et al., 2015; Bojar et al., 2016; Bojar et al., 2017).

To provide awareness of errors in mt originating from src, attention mechanisms (Bahdanau et al., 2015) allow modeling of non-local dependencies in the input or output sequences, and importantly also global dependencies between them (in our case src, mt and pe). The *transformer* architecture (Vaswani et al., 2017) is built solely upon such attention mechanisms completely replacing recurrence and convolutions. The transformer uses positional encoding to encode the input and output sequences, and computes both self- and cross-attention through so-called multi-head attentions, which can be easily parallelized. Multi-head attention allows to jointly attend to information at different positions from different representation subspaces, e.g. utilizing and combining information from src, mt, and pe.

In this paper, we present a multi-encoder based neural APE architecture called *transference*. Our model contains a source encoder which encodes src information, a second encoder ($enc_{src \rightarrow mt}$) which takes the encoded representation from the source encoder (enc_{src}), combines this with the self-attention-based encoding of mt (enc_{mt}), and prepares a representation for the decoder (dec_{pe}) via cross-attention. Our second encoder ($enc_{src \rightarrow mt}$) can also be viewed as a standard transformer decoding block, however, without masking, which acts as an encoder. We thus recombine the different blocks of the transformer architecture and repurpose them for the APE task in a simple yet effective way. The suggested architecture is inspired by the two-step approach professional translators tend to use during post-editing: first, the source segment is compared to the corresponding translation suggestion (similar to what our $enc_{src \rightarrow mt}$ is doing), then corrections to the MT output are applied based on the encountered errors (in the same way that our dec_{pe} uses the encoded representation of $enc_{src \rightarrow mt}$ to produce the final translation).

The paper makes the following contributions: (i) we propose a multi-encoder model for APE that consists only of standard transformer encoding and decoding blocks, (ii) by using a mix of self- and cross-attention we provide a representation of both src and mt for the decoder, allowing it to better capture errors in mt originating from src; this advances Junczys-Dowmunt and Grundkiewicz (2018) – the WMT 2018 best system ($wmt18^{smt}_{best}$) in terms of BLEU and TER, (iii), we analyze the effect of varying the number of encoder and decoder layers (Domhan, 2018), indicating that the encoders contribute more than decoders in neural APE, and (iv) we present and evaluate an APE architecture inspired by a two-step approach professional translators often use during post-editing.

In comparison to the shared task system description paper (Pal et al., 2019), this paper (i) provides more detailed explanations and reformation of different components of the *transference* architecture, (ii) compares it to a single encoder based transformer architecture where only mt or src concatenated with mt are used as an input, (iii) analyzes results when swapping mt and src in the multi-encoder setup, and (iv) investigates the importance of encoder and decoder by varying the amount of layers.

The rest of the paper is organized as follows. In §2, we survey existing literature on APE. In §3, we describe the *multi-encoder* architecture. §4 describes our experimental setup. §5 reports the results of our approach against a number of baselines. Finally, §6 concludes the paper with future directions.

2 Related Research

Recent advances in APE research are directed towards neural APE, which was first proposed by Pal et al. (2016b) and Junczys-Dowmunt and Grundkiewicz (2016) for the single-source APE scenario which does not consider src, i.e. $mt \rightarrow pe$. Junczys-Dowmunt and Grundkiewicz (2016) also generated a large synthetic training dataset, which we also use as additional training data.

Exploiting source information as an additional input can help neural APE to disambiguate corrections applied at each time step; this naturally leads to multi-source APE ($\{src, mt\} \rightarrow pe$). 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 few approaches to multi-source neural APE were proposed in the WMT 2017 APE shared task. Junczys-Dowmunt and Grundkiewicz (2017) combine both mt and src in a single neural architecture, exploring different combinations of attention mechanisms including soft attention and hard monotonic attention. Chatterjee et al. (2017) built upon the two-encoder architecture of multi-source models (Libovický et al., 2016) by means of concatenating both weighted contexts of encoded src and mt. Varis and Bojar (2017) compared two multi-source models, one using a single encoder with the concatenation of src and mt sentences, and a second one using two character-level encoders for mt and src along with a character-level decoder.

In the WMT 2018 APE shared task, several adaptations of the transformer architecture were presented for multi-source APE. Pal et al. (2018) introduced a joint encoder that attends over a combination of the two encoded sequences from mt and src. Tebbifakhr et al. (2018), the NMT-subtask winner of WMT 2018 ($wmt18^{nmt}_{best}$), employed sequence-level loss functions in order to avoid exposure bias during training and to be consistent with the automatic evaluation metrics. Shin and Lee (2018) proposed a multi-source transformer where on the decoder side, they added two additional multi-head attention layers for $src \rightarrow mt$ and $src \rightarrow pe$. Thereafter 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 APE PBSMT-subtask winner of WMT 2018 ($wmt18^{smt}_{best}$) (Junczys-Dowmunt and Grundkiewicz, 2018) also presented another transformer-based multi-source APE which uses two encoders and stacks an additional cross-attention component for $src \rightarrow pe$ above the previous cross-attention for $mt \rightarrow pe$.

In contrast to other multi-encoder based approaches and Libovický et al. (2018)'s approach, where the authors focused on cross-attention of two encoders with respect to the decoder within the transformer architecture, we propose a novel architecture where the second encoder block is similar to the transformer decoder block but without masking.

In the latest edition of WMT (2019), the submissions are mostly multi-source models extending the transformer implementation (Pal et al., 2019; Lee et al., 2019; Xu et al., 2019) and adapting BERT (Devlin et al., 2018) to the transformer-based framework (Lopes et al., 2019). The winner system (Lopes et al., 2019) ($wmt19^{nmt}_{best}$) uses a single pre-trained BERT encoder that receives both the source src and mt strings and applies a BERT-based encoder-decoder model. Additionally, they add a conservativeness penalty factor during beam decoding to avoid over-corrections in APE.

Our method outperforms the WMT 2016, 2017, and 2018 winners by 1 BLEU point, and yields comparable performance to the WMT 2019 winner, however, without using a BERT-based architecture.

3 The Transference Multi-Encoder Transformer for APE

We propose a multi-source transformer model called *transference* ($\{src, mt\}_{tr} \rightarrow pe$, 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, $enc_{src\rightarrow mt}$, makes use of the first encoder enc_{src} and a sub-encoder enc_{mt} for considering src and mt. Here, the enc_{src} encoder and the dec_{pe} decoder are equivalent to the original transformer for neural MT (Vaswani et al., 2017). Our $enc_{src\rightarrow mt}$ follows an architecture similar to the transformer's decoder, the difference being that multihead self-attention is not masked to process mt.

The self-attended encoder for src, $\mathbf{s} = (s_1, s_2, \ldots, s_k)$, returns a sequence of continuous representations, enc_{src}, and the second self-attended sub-encoder for mt, $\mathbf{m} = (m_1, m_2, \ldots, m_l)$, returns another sequence of continuous representations, enc_{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. To do this the internal enc_{mt} representation performs cross-attention over enc_{src} and prepares a final representation ($enc_{src\rightarrow mt}$) for the decoder (dec_{pe}). The decoder then generates the pe output in sequence, $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, one word at a time from left to right by attending to previously generated words as well as the final representations ($enc_{src\rightarrow mt}$) generated by the encoder.

To summarize, 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.

Our proposed approach differs from the WMT 2018 PBSMT winner system ($wmt18^{smt}_{best}$) in several ways: (i) we use the original transformer's decoder without modifications; (ii) one of our encoder blocks ($enc_{src\rightarrow mt}$) is identical to the transformer's decoder block but uses no masking in the self-attention layer, thus having one self-attention layer and an additional cross-attention for $src \rightarrow mt$; and (iii) in the decoder layer, the cross-attention is performed between the encoded representation from $enc_{src\rightarrow mt}$ and pe. Moreover, placing a cross-attention network within the $enc_{src\rightarrow mt}$ sub-layer rather than the dec_{pe} sub-layer as in $wmt18^{smt}_{best}$, during inference, $enc_{src\rightarrow mt}$ is forward propagated only once instead of multiple times i.e., once per decoding step.

Our approach also differs from the WMT 2018 NMT winner system: (i) $wmt18^{nmt}_{best}$ concatenates the encoded representation of two encoders and passes it as the key to the attention layer of the decoder, and (ii), the system additionally employs sequence-level loss functions based on maximum likelihood estimation and minimum risk training in order to avoid exposure bias during training.

Comparing with $wmt19^{nmt}_{best}$, the winner system of WMT 2019 uses a pre-trained deep bidirectional transformer (multilingual BERT) (Devlin et al., 2018), while our model does not.

The main intuition is that our $enc_{src \rightarrow mt}$ attends over the src and mt and informs the pe to better capture, process, and share information between src-mt-pe, which efficiently models error patterns and the corresponding corrections. Our model performs better than past transformer-based approaches and similar to the BERT-based approach ($wmt19^{nmt}_{best}$) without adding the overhead of the pre-trained model, as the experiment section will show.

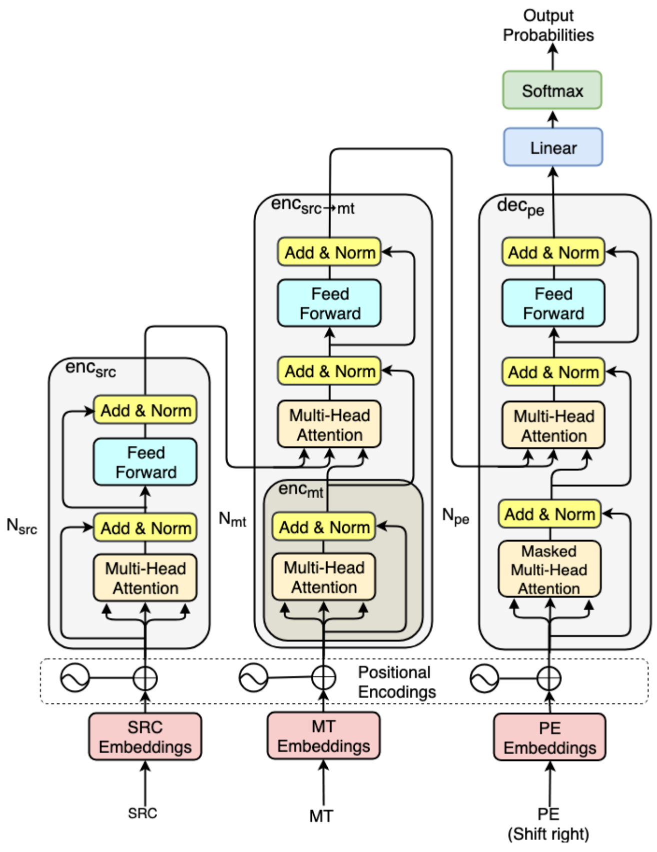

Figure 1: The *transference* model architecture for APE ($\{src, mt\}_{tr} \rightarrow pe$).

4 Experiments

We explore our approach on both APE sub-tasks of WMT 2018, where the black box MT (we refer as 1st-stage MT) system to which APE is applied is either a phrase-based statistical machine translation (PBSMT) or a neural machine translation (NMT) model.

For the PBSMT task, we compare against four baselines: the **raw SMT** output provided by the 1st-stage PBSMT, the best-performing systems from WMT APE 2018 ($\mathbf{wmt18^{smt}_{best}}$), which are a single model and an ensemble model by Junczys-Dowmunt and Grundkiewicz (2018), as well as a transformer directly translating from src to pe (**Transformer** ($\mathbf{src} \rightarrow \mathbf{pe}$)), thus performing translation instead of APE. We evaluate the systems using BLEU (Papineni et al., 2002) and TER (Snover et al., 2006).

For the NMT task, we consider three baselines: the **raw NMT** output provided by the 1st-stage NMT system, the best-performing system from the WMT 2018 ($\mathbf{wmt18^{nmt}_{best}}$) (Tebbifakhr et al., 2018) and WMT 2019 ($\mathbf{wmt19^{nmt}_{best}}$) (Lopes et al., 2019) NMT APE task.

Apart from the multi-encoder *transference* architecture described above ($\{src, mt\}_{tr} \rightarrow pe$) and ensembling of this architecture, two simpler versions are also analyzed: first, a 'mono-lingual' ($\mathbf{mt} \rightarrow \mathbf{pe}$) APE model using only parallel mt–pe data and therefore only a single encoder, and second, an identical single-encoder architecture, however, using the concatenated src and mt text as input ($\{\mathbf{src} + \mathbf{mt}\} \rightarrow \mathbf{pe}$) (Niehues et al., 2016).

4.1 Data

For our experiments, we use the English–German WMT 2016 (Bojar et al., 2016), 2017 (Bojar et al., 2017), 2018 (Chatterjee et al., 2018) and 2019 (Chatterjee et al., 2019) APE task data. All these 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 MT system, and the corresponding human-post-edited version (pe). The sizes of the datasets (train; dev; test), in terms of number of sentences, are (12,000; 1,000; 2,000), (11,000; 0; 2,000), and (13,442; 1,000; 1,023), for the 2016 PBSMT, the 2017 PBSMT, and the 2018 NMT data, respectively. The 2019 version of the APE dataset released in WMT is the same as the WMT 2018 NMT data. It is to be noted that for WMT 2018, we carried out experiments only for the NMT sub-task and ignored the data for the PBSMT task.

Since the WMT APE datasets are small in size, we use 'artificial training data' (Junczys-Dowmunt and Grundkiewicz, 2016) containing 4.5M sentences as additional resources, 4M of which are weakly similar to the WMT 2016 training data, while 500K are very similar according to TER statistics.

For experimenting on the NMT data, we additionally use the synthetic eScape APE corpus (Negri et al., 2018), consisting of $\sim$7M triples. For cleaning this noisy eScape dataset containing many unrelated language words (e.g. Chinese), (i) we use the cleaning process described in Tebbifakhr et al. (2018), and (ii) we use the Moses (Koehn et al., 2007) corpus cleaning scripts with minimum and maximum number of tokens set to 1 and 100, respectively. After cleaning, we perform punctuation normalization, and then use the Moses tokenizer (Koehn et al., 2007) to tokenize the eScape corpus with 'no-escape' option. Finally, we apply true-casing. The cleaned version of the eScape corpus contains $\sim$6.5M triplets.

4.2 Experiment Setup

To build models for the PBSMT tasks from 2016 and 2017, we first train a generic APE model using all the training data (4M + 500K + 12K + 11K) described in Section 4.1. Afterwards, we fine-tune the trained model using the 500K artificial and 23K (12K + 11K) real PE training data. We use the WMT 2016 development data (dev2016) containing 1,000 triplets to validate the models during training. To test our system performance, we use the WMT 2016 and 2017 test data (test2016, test2017) as two sub-experiments, each containing 2,000 triplets (src, mt and pe). We compare the performance of our system with the four different baseline systems described above: raw MT, $wmt18^{smt}_{best}$ single and ensemble, as well as transformer ($src \rightarrow pe$).

Additionally, we check the performance of our model on the WMT 2018 NMT APE task (where unlike in previous tasks, the 1st-stage MT system is provided by NMT): for this, we explore two experimental setups: (i) we use the PBSMT task's APE model as a generic model which is then fine-tuned to a subset (12k) of the NMT data ($\{src, mt\}^{nmt}_{tr} \rightarrow pe^{generic,smt}$). One should note that it has been argued that the inclusion of SMT-specific data could be harmful when training NMT APE models (Junczys-Dowmunt and Grundkiewicz, 2018). (ii), we train a completely new generic model on the cleaned eScape data ($\sim$6.5M) along with a subset (12K) of the original training data released for the NMT task ($\{src, mt\}^{nmt}_{tr} \rightarrow pe^{generic,nmt}$). The aforementioned 12K NMT data are the first 12K of the overall 13.4K NMT data. The remaining 1.4K are used as validation data. The released development set (dev2018) is used as test data for our experiment, alongside the test2018, for which we could only obtain results for a few models by the WMT 2019 task organizers. We also explore an additional fine-tuning step of $\{src, mt\}^{nmt}_{tr} \rightarrow pe^{generic,nmt}$ towards the 12K NMT data (called $\{src, mt\}^{nmt}_{tr} \rightarrow pe^{ft}$), and a model averaging the 8 best checkpoints of $\{src, mt\}^{nmt}_{tr} \rightarrow pe^{ft}$, which we call $\{src, mt\}^{nmt}_{tr} \rightarrow pe^{ft}_{avg}$.

During post-editing, professional translators have to understand the source, and analyze if the MT correctly represents the source, which corresponds to our enc_{src} and $enc_{src \rightarrow mt}$. To investigate whether following this realistic understanding of the post-editing process is beneficial, we compare the model to a version with swapped inputs (mt, src), called $\{mt, src\}^{smt}_{tr} \rightarrow pe^{generic}$. We carried out an experiment with the PBSMT task's APE dataset. Moreover, we fine-tune the $\{mt, src\}^{smt}_{tr} \rightarrow pe^{generic}$ model with 500K artificial and 23K real PE training data and compare the fine-tuned model ($\{mt, src\}^{smt}_{tr} \rightarrow pe^{ft}$) with $\{src, mt\}^{smt}_{tr} \rightarrow pe^{ft}$.

Last, we analyze the importance of our second encoder ($enc_{src \rightarrow mt}$), compared to the source encoder

(enc_{src}) and the decoder (dec_{pe}), by reducing and expanding the amount of layers in the encoders and the decoder. Our standard setup, which we use for fine-tuning, ensembling etc., is fixed to 6-6-6 for N_{src}-N_{mt}-N_{pe} (cf. Figure 1). We investigate what happens in terms of APE performance if we change this setting to 6-6-4 and 6-4-6.

To handle out-of-vocabulary words and 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 (German) and src is a close language (English), they naturally share a good fraction of BPE tokens, which reduces the vocabulary size to 28k. We implemented our approach based on the Neutron implementation of the Transformer (Xu and Liu, 2019)[1].

4.3 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 (except for the layer experiment) followed by layer normalization. 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. 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.

Exp. no.	Models	test2016		test2017	
		BLEU ↑	TER ↓	BLEU ↑	TER ↓
Baselines					
1.1	Raw SMT	62.11	24.76	62.49	24.48
1.2	Transformer ($src \rightarrow pe$)	56.59 (-5.52)	29.97 (+5.21)	53.06 (-9.43)	32.20 (+7.72)
1.3	$wmt18^{smt}_{best}$ (single)	70.86 (+8.75)	18.92 (-5.84)	69.72 (+7.23)	19.49 (-4.99)
1.4	$wmt18^{smt}_{best}$ (x4)	**71.04** (+8.93)	**18.86** (-5.9)	**70.46** (+7.97)	**19.03** (-5.45)
Baselines: Retrained $wmt18^{smt}_{best}$ with our experimental setup					
1.5	$wmt18^{smt,generic}_{best}$ (single)	69.14 (+7.03)	20.41 (-4.35)	68.14 (+5.65)	20.98 (-3.5)
1.6	$wmt18^{smt,ft}_{best}$ (single)	70.12 (+8.01)	19.84 (-4.92)	69.16 (+6.67)	20.34 (-4.14)
General models trained on 23K+4.5M data					
2.1	$mt \rightarrow pe$	67.70 (+5.59)	21.90 (-2.86)	66.91 (+4.42)	22.32 (-2.16)
2.2	$\{src + mt\} \rightarrow pe$	69.32 (+7.21)	20.27 (-4.49)	68.26 (+5.77)	20.90 (-3.58)
2.3	$\{src, mt\}^{smt}_{tr} \rightarrow pe$	70.46 (+8.35)	19.21 (-5.55)	70.05 (+7.56)	19.46 (-5.02)
2.4	$\{mt, src\}^{smt}_{tr} \rightarrow pe$	70.26 (+8.15)	19.34 (-5.42)	69.34 (+6.85)	20.05 (-4.43)
Fine-tuning Exp. 2 models with 23K+500K data					
3.1	$mt \rightarrow pe$	68.43 (+6.32)	21.29 (-3.47)	67.78 (+5.29)	21.63 (-2.85)
3.2	$\{src + mt\} \rightarrow pe$	69.87 (+7.76)	19.94 (-4.82)	68.57 (+6.08)	20.68 (-3.8)
3.3	$\{src, mt\}^{smt}_{tr} \rightarrow pe^{ft}$	71.05 (+8.94)	19.05 (-5.71)	70.33 (+7.84)	19.23 (-5.25)
3.4	$\{mt, src\}^{smt}_{tr} \rightarrow pe^{ft}$	70.26 (+8.15)	19.40 (-5.36)	69.31 (+6.82)	19.91 (-4.57)
4.1	Exp3.3$^{smt}_{ens4ckpt}$	**71.59** (+9.48)	**18.78** (-5.98)	**70.89** (+8.4)	**18.91** (-5.57)
4.2	$ensemble^{smt}(x3)$	**72.19** (+10.08)	**18.39** (-6.37)	**71.58** (+9.09)	**18.58** (-5.90)
$\{src, mt\}^{smt}_{tr} \rightarrow pe$ **with different layer size**					
5.1	$\{src, mt\}^{smt}_{tr} \rightarrow pe$ (6-6-4)	70.85 (+8.74)	19.00 (-5.76)	69.82 (+7.33)	19.67 (-4.81)
5.2	$\{src, mt\}^{smt}_{tr} \rightarrow pe$ (6-4-6)	69.93 (+7.82)	19.70 (-5.06)	69.61 (+7.12)	19.68 (-4.8)

Table 1: Evaluation results on the WMT APE test set 2016, and test set 2017 for the PBSMT task; ($\pm X$) value is the improvement over Raw SMT. The last section of the table shows the impact of increasing and decreasing the depth of the encoders and the decoder. Our *transference* model is 1.07 times faster than the dual source model ($wmt18^{smt}_{best}$) for post-editing the testset of 2000 sentences.

[1] `https://github.com/anoidgit/transformer`.

5 Results

The results of our four models, ***single-source*** ($\mathbf{mt} \rightarrow \mathbf{pe}$), ***multi-source single encoder*** ($\{\mathbf{src} + \mathbf{pe}\} \rightarrow \mathbf{pe}$), ***transference*** model ($\{\mathbf{src}, \mathbf{mt}\}^{\mathbf{smt}}_{\mathbf{tr}} \rightarrow \mathbf{pe}$, $\{\mathbf{mt}, \mathbf{src}\}^{\mathbf{smt}}_{\mathbf{tr}} \rightarrow \mathbf{pe}$), and ***ensemble***, in comparison to the four baselines, ***raw SMT***, $\mathbf{wmt18^{smt}_{best}}$ (Junczys-Dowmunt and Grundkiewicz, 2018) single and ensemble, as well as ***Transformer*** ($\mathbf{src} \rightarrow \mathbf{pe}$), are presented in Table 1 for test2016 and test2017. Table 2 reports the results obtained by our **transference** model ($\{\mathbf{src}, \mathbf{mt}\}^{\mathbf{nmt}}_{\mathbf{tr}} \rightarrow \mathbf{pe}$) on the WMT 2018, 2019 NMT data for dev2018 (which we use as a test set) and test2018/2019 (when testset was available), compared to the baselines ***raw NMT***, $\mathbf{wmt18^{nmt}_{best}}$, and $\mathbf{wmt19^{nmt}_{best}}$.

5.1 Baselines

The **raw SMT** output in Table 1 is a strong black-box PBSMT system (i.e., 1st-stage MT). We report its performance observed with respect to the ground truth (pe), i.e., the post-edited version of mt. The original PBSMT system scores over 62 BLEU points and below 25 TER on test2016 and test2017.

Using a **Transformer** ($src \rightarrow pe$), we test if APE is really useful, or if potential gains are only achieved due to the good performance of the transformer architecture. While we cannot do a full training of the transformer on the data that the raw MT engine was trained on due to the unavailability of the data, we use our PE datasets in an equivalent experimental setup as for all other models. The results of this system (Exp. 1.2 in Table 1) show that the performance is actually lower across both test sets, -5.52/-9.43 absolute points in BLEU and +5.21/+7.72 absolute in TER, compared to the raw SMT baseline.

We report four results from $\mathbf{wmt18^{smt}_{best}}$, (i) $wmt18^{smt}_{best}$ ($single$), which is the core multi-encoder implementation without ensembling but with checkpoint averaging, (ii) $wmt18^{smt}_{best}$ ($x4$) which is an ensemble of four identical 'single' models trained with different random initializations. The results of $wmt18^{smt}_{best}$ ($single$) and $wmt18^{smt}_{best}$ ($x4$) (Exp. 1.3 and 1.4) reported in Table 1 are from Junczys-Dowmunt and Grundkiewicz (2018). Since their training procedure slightly differs from ours, we also trained the $wmt18^{smt}_{best}$ system using exactly our experimental setup in order to make a fair comparison. This yields the baselines (iii) $wmt18^{smt,generic}_{best}$ ($single$) (Exp. 1.5), which is similar to $wmt18^{smt}_{best}$ ($single$), however, the training parameters and data are kept in line with our *transference* general model (Exp. 2.3) and (iv) $wmt18^{smt,ft}_{best}$ ($single$) (Exp. 1.6), which is also trained maintaining the equivalent experimental setup compared to the fine tuned version of the *transference* general model (Exp. 3.3). Compared to both raw SMT and transformer ($src \rightarrow pe$) we see strong improvements for this state-of-the-art model, with BLEU scores of at least 68.14 and TER scores of at most 20.98 across the PBSMT testsets. $wmt18^{smt}_{best}$, however, performs better in its original setup (Exp. 1.3 and 1.4) compared to our experimental setup (Exp. 1.5 and 1.6).

The results on the WMT 2018 and 2019 NMT datasets (dev2018 and test2018) are presented in Table 2. The *raw NMT* system serves as one baseline against which we compare the performance of the different models. We evaluate the system hypotheses with respect to the ground truth (pe), i.e., the post-edited version of mt. The baseline original NMT system scores 76.76 BLEU points and 15.08 TER on dev2018, and 74.73 BLEU points and 16.80 TER on test2018.

5.2 Single-Encoder Transformer for APE

The two transformer architectures $\mathbf{mt} \rightarrow \mathbf{pe}$ and $\{\mathbf{src} + \mathbf{mt}\} \rightarrow \mathbf{pe}$ use only a single encoder. Table 1 shows that $\mathbf{mt} \rightarrow \mathbf{pe}$ (Exp. 2.1) provides better performance (+4.42 absolute BLEU on test2017) compared to the original SMT, while $\{\mathbf{src} + \mathbf{mt}\} \rightarrow \mathbf{pe}$ (Exp. 2.2) provides further improvements by additionally using the src information. $\{\mathbf{src} + \mathbf{mt}\} \rightarrow \mathbf{pe}$ improves over $\mathbf{mt} \rightarrow \mathbf{pe}$ by +1.62/+1.35 absolute BLEU points on test2016/test2017. After fine-tuning, both single encoder transformers (Exp. 3.1 and 3.2 in Table 1) show further improvements, +0.87 and +0.31 absolute BLEU points, respectively, for test2017 and a similar improvement for test2016.

5.3 Transference Transformer for APE

In contrast to the two models above, our *transference* architecture uses multiple encoders. The fine-tuned version of the $\{src, mt\}^{smt}_{tr} \rightarrow pe$ model (Exp. 3.3 in Table 1) outperforms $wmt18^{smt}_{best}$ (single)

(Exp. 1.3) in BLEU on both test sets, however, the TER score for test2016 increases. When ensembling the 4 best checkpoints of our $\{src, mt\}_{tr}^{smt} \rightarrow pe$ model (Exp. 4.1), the result beats the $wmt18_{best}^{smt}$ (x4) system, which is an ensemble of four different randomly initialized $wmt18_{best}^{smt}$ (single) systems. Our $\mathbf{ensemble^{smt}(x3)}$ combines two $\{src, mt\}_{tr}^{smt} \rightarrow pe$ (Exp. 2.3) models initialized with different random weights with the ensemble of the fine-tuned transference model Exp3.$3_{ens4ckpt}^{smt}$(Exp. 4.1). This ensemble provides the best results for all datasets, providing roughly +1 BLEU point and -0.5 TER when comparing against $wmt18_{best}^{smt}$ (x4). In terms of the number of parameters, $wmt18_{best}^{smt}$ and our $\{src, mt\}_{tr}^{smt} \rightarrow pe$ model are the same. Moreover, our $\{src, mt\}_{tr}^{smt} \rightarrow pe$ model uses a single multi-head cross-attention in the decoder sub-layer, compared to two multi-head cross-attention mechanisms in $wmt18_{best}^{smt}$, therefore our model requires less inference time. Furthermore, using more non-autoregressive encoder layers with fewer autoregressive decoder layers can significantly accelerate the inference (Xu et al., 2020), instead of aggregating src and mt with the autoregressive pe decoder (Junczys-Dowmunt and Grundkiewicz, 2018), our approach that aggregates src and mt with the non-autoregressive mt encoder is significantly faster than the $wmt18_{best}^{smt}$ in inference, which is of practical value.

Additionally we compare our $\{src, mt\}_{tr}^{smt} \rightarrow pe$ model with $\{mt, src\}_{tr}^{smt} \rightarrow pe$, where we reverse the input order i.e., enc_1 and enc_2 take mt and src, respectively, as input. Exp. 2.4 and Exp. 3.4 report $\{mt, src\}_{tr}^{smt} \rightarrow pe$ and $\{mt, src\}_{tr}^{smt} \rightarrow pe^{ft}$ respectively, which performed slightly worse than $\{src, mt\}_{tr}^{smt} \rightarrow pe$ and $\{src, mt\}_{tr}^{smt} \rightarrow pe^{ft}$. Surprisingly, fine-tuning does not help $\{mt, src\}_{tr}^{smt} \rightarrow pe^{ft}$ for the testset 2016, however, in case of testset 2017, fine-tuning shows small gain in performance. Moreover, the performance gain in fine-tuning for the case of $\{src, mt\}_{tr}^{smt} \rightarrow pe^{ft}$ over $\{src, mt\}_{tr}^{smt} \rightarrow pe$ is considerably stronger than the performance gain for $\{mt, src\}_{tr}^{smt} \rightarrow pe^{ft}$ over $\{mt, src\}_{tr}^{smt} \rightarrow pe$. Empirically, this manifests our hypothesis that our model ($\{src, mt\}_{tr}^{smt} \rightarrow pe$) follows human post-editors' two-step approach: first, the source segment is compared to the corresponding translation suggestion, then corrections to the MT output are applied based on the encountered errors.

Exp. no.	Models	dev2018		test2018	
		BLEU ↑	TER ↓	BLEU ↑	TER ↓
6.1	Raw NMT	76.76	15.08	74.73	16.80
6.2	$wmt18_{best}^{nmt}$	**77.74 (+0.98)**	14.78 (-0.30)	75.53 (+0.80)	16.46 (-0.34)
6.3	$wmt19_{best}^{nmt}$	-	-	**75.96 (+1.23)**	**16.06 (-0.74)**
Fine-tuning Exp. 3.3 on 12k NMT data					
7	$\{src, mt\}_{tr}^{nmt} \rightarrow pe^{generic,smt}$	77.09 (+0.33)	14.94 (-0.14)	-	-
Transference model trained on eScape+ 12k NMT data					
8	$\{src, mt\}_{tr}^{nmt} \rightarrow pe^{generic,nmt}$	77.25 (+0.49)	14.87 (-0.21)	-	-
Fine-tuning model 8 on 12k NMT data					
9	$\{src, mt\}_{tr}^{nmt} \rightarrow pe^{ft}$	77.39 (+0.63)	14.71 (-0.37)	-	-
Averaging 8 checkpoints of Exp. 9					
10	$\{src, mt\}_{tr}^{nmt} \rightarrow pe_{avg}^{ft}$	77.67 (+0.91)	**14.52 (-0.56)**	75.75 (+1.02)	16.15 (-0.69)

Table 2: Evaluation results on the WMT APE 2018 development set for the NMT task (Exp. 6 and Exp. 10 results were obtained by the WMT 2019 task organizers).($\pm X$) value is the improvement over Raw NMT.

For the WMT 2018 NMT data we first test our $\{src, mt\}_{tr}^{nmt} \rightarrow pe^{generic,smt}$ model, which is the model from Exp. 3.3 fine-tuned towards NMT data as described in Section 4.2. Table 2 shows that our PBSMT APE model fine-tuned towards NMT (Exp. 7) can even slightly improve over the already very strong NMT system by about +0.3 BLEU and -0.1 TER, although these improvements are not statistically significant.

The overall results improve when we train our model on eScape and NMT data instead of using the PBSMT model as a basis. Our proposed generic *transference* model (Exp. 8, $\{src, mt\}_{tr}^{nmt} \rightarrow pe^{generic,nmt}$) shows statistically significant improvements in terms of BLEU and TER compared to the baseline even before fine-tuning, and further improvements after fine-tuning (Exp. 9, $\{src, mt\}_{tr}^{nmt} \rightarrow pe^{ft}$). Finally, after averaging the 8 best checkpoints, our $\{src, mt\}_{tr}^{nmt} \rightarrow pe_{avg}^{ft}$ model (Exp. 10) also shows consistent improvements in comparison to the baseline and other experimental setups. Overall our fine-tuned model averaging the 8 best checkpoints achieves +1.02 absolute BLEU points and -0.69 absolute TER improvements over the baseline on test2018. Table 2 also shows the performance of our model compared to the winner system of WMT 2018 ($wmt18^{nmt}_{best}$) for the NMT task (Tebbifakhr et al., 2018). $wmt18^{nmt}_{best}$ scores 14.78 in TER and 77.74 in BLEU on the dev2018 and 16.46 in TER and 75.53 in BLEU on the test2018. In comparison to $wmt18^{nmt}_{best}$, our model (Exp. 10) achieves better scores in TER on both the dev2018 and test2018, however, in terms of BLEU our model scores slightly lower for dev2018, while some improvements are achieved on test2018. Compared to $wmt19^{nmt}_{best}$ (Exp. 6.3), our model scores slightly lower, however, the performance loss is not statistically significant. It is to be noted that the training strategy in $wmt19^{nmt}_{best}$ is different – (i) they used their own synthetic corpus prepared using the parallel data provided by the Quality Estimation shared task[2], (ii) they oversampled the APE training data 20 times, and (iii) they applied multilingual BERT.

The number of layers (N_{src}-N_{mt}-N_{pe}) in all encoders and the decoder for these results is fixed to 6-6-6. In Exp. 5.1, and 5.2 in Table 1, we see the results of changing this setting to 6-6-4 and 6-4-6. This can be compared to the results of Exp. 2.3, since no fine-tuning or ensembling was performed for these three experiments. Exp. 5.1 shows that decreasing the number of layers on the decoder side does not hurt the performance. In fact, in the case of test2016, we got some improvement, while for test2017, the scores got slightly worse. In contrast, reducing the $enc_{src\rightarrow mt}$ encoder block's depth (Exp. 5.2) does indeed reduce the performance for all four scores, showing the importance of this second encoder.

5.4 Discussion

The proposed multi-encoder based transformer architecture ($\{src, mt\}^{smt}_{tr} \rightarrow pe$, Exp. 2.3) shows slightly worse results than $wmt18^{smt}_{best}$ (single) (Exp. 1.3) before fine-tuning, and roughly similar results after fine-tuning (Exp. 3.3). After ensembling, however, our *transference* model (Exp. 4.2) shows consistent improvements when comparing against the best baseline ensemble $wmt18^{smt}_{best}$ (x4) (Exp. 1.4). Due to the unavailability of the sentence-level scores of $wmt18^{smt}_{best}$ (x4), we could not test if the improvements (roughly +1 BLEU, -0.5 TER) are statistically significant. Interestingly, our approach of taking the model optimized for PBSMT and fine-tuning it to the NMT task (Exp. 7) does not hurt the performance as was reported in the previous literature (Junczys-Dowmunt and Grundkiewicz, 2018). In contrast, some small, albeit statistically insignificant improvements over the raw NMT baseline were achieved. When we train the *transference* architecture directly for the NMT task (Exp. 8), we get slightly better and statistically significant improvements compared to raw NMT. Fine-tuning this NMT model further towards the actual NMT data (Exp. 9), as well as performing checkpoint averaging using the 8 best checkpoints improves the results even further. Compared to $wmt18^{smt}_{best}$ and $wmt19^{nmt}_{best}$, our architecture is simpler, faster during inference, it follows the two-step approach of professional post-editors, and has no additional overhead like BERT.

The reasons for the effectiveness of our approach can be summarized 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 $wmt18^{smt}_{best}$, the dec_{pe} accesses representations from src and mt independently, first using the representation from mt and then using that of src. (2) The position-wise feed-forward layer in our $enc_{src\rightarrow mt}$ of our model requires processing information from two attention modules, while in the case of $wmt18^{smt}_{best}$, the position-wise feed-forward layer in dec_{tgt} needs to process information from three attention modules, which may increase the learning difficulty of the feed-forward layer. (3) 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 than those achieved by attending over src directly.

Evaluating the influence of the depth of our encoders and decoder show that while the decoder depth

[2]http://www.statmt.org/wmt19/qe-task.html

appears to have limited importance, reducing the encoder depth indeed hurts performance which is in line with Domhan (2018).

6 Conclusions

In this paper, we presented a multi-encoder transformer-based APE model that repurposes the standard transformer blocks in a simple and effective way for the APE task: first, our *transference* architecture uses a transformer encoder block for src, followed by a decoder block without masking on mt that effectively acts as a second encoder combining $src \rightarrow mt$, and feeds this representation into a final decoder block generating pe. The proposed model outperforms the best-performing system of WMT 2018 on the test2016, test2017, dev2018, and test2018 data. Moreover, our model is on par with but simpler than WMT 2019 best system since our model does not apply BERT or any conservative factor during inference.

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. We also show that the encoder plays a more pivotal role than the decoder in transformer-based APE, which could also be the case for transformer-based generation tasks in general. Our architecture is generic and can be used for any multi-source task, e.g., (i) Multi-source Translation (ii) document translation to model the associated context during translation, (iii) Question Generation to generate question from given passage and a short answer text, (iv) Question Answering task from given passage and question text, (v) Summarization, etc.

Acknowledgements

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). Hongfei Xu is supported by China Scholarship Council ([2018]3101, 201807040056). 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.

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, September. 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, August. Association for Computational Linguistics.

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, September. 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, September. 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, October. Association for Computational Linguistics.

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the wmt 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 13–30, Florence, Italy, August. 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*.

Tobias Domhan. 2018. How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1799–1808, Melbourne, Australia, July. 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.

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, September. 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, October. 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.

WonKee Lee, Jaehun Shin, and Jong-Hyeok Lee. 2019. Transformer-based automatic post-editing model with joint encoder and multi-source attention of decoder. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 114–119, Florence, Italy, August. Association for Computational Linguistics.

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, August. Association for Computational Linguistics.

Jindřich Libovický, Jindřich Helcl, and David Mareček. 2018. Input combination strategies for multi-source transformer decoder. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 253–260, Belgium, Brussels, October. Association for Computational Linguistics.

António V. Lopes, M. Amin Farajian, Gonçalo M. Correia, Jonay Trénous, and André F. T. Martins. 2019. Unbabel's submission to the wmt2019 ape shared task: Bert-based encoder-decoder for automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 120–125, Florence, Italy, August. 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 Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018. 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, December. The COLING 2016 Organizing Committee.

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, August. Association for Computational Linguistics.

Santanu Pal, Nico Herbig, Antonio Krüger, 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, October. Association for Computational Linguistics.

Santanu Pal, Hongfei Xu, Nico Herbig, Antonio Krüger, and Josef van Genabith. 2019. USAAR-DFKI – The Transference Architecture for English–German Automatic Post-Editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 126–133, Florence, Italy, August. 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.

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, October. 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, October. 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, September. 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.

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

Hongfei Xu, Qiuhui Liu, and Josef van Genabith. 2019. Uds submission for the wmt 19 automatic post-editing task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 147–152, Florence, Italy, August. Association for Computational Linguistics.

Hongfei Xu, Josef van Genabith, Deyi Xiong, and Qiuhui Liu. 2020. Analyzing Word Translation of Transformer Layers. *arXiv preprint arXiv:2003.09586*, March.