Transformer-based Automatic Post-Editing Model with Joint Encoder and Multi-source Attention of Decoder

WonKee Lee*, Jaehun Shin*, Jong-hyeok Lee
Department of Computer Science and Engineering,
Pohang University of Science and Technology (POSTECH), Republic of Korea
{wklee, jaehun.shin, jhlee}@postech.ac.kr

Abstract

This paper describes POSTECH's submission to the WMT 2019 shared task on Automatic Post-Editing (APE). In this paper, we propose a new multi-source APE model by extending Transformer. The main contributions of our study are that we 1) reconstruct the encoder to generate a joint representation of translation (*mt*) and its *src* context, in addition to the conventional *src* encoding and 2) suggest two types of multi-source attention layers to compute attention between two outputs of the encoder and the decoder state in the decoder. Furthermore, we train our model by applying various teacher-forcing ratios to alleviate exposure bias. Finally, we adopt the ensemble technique across variations of our model. Experiments on the WMT19 English-German APE data set show improvements in terms of both TER and BLEU scores over the baseline. Our primary submission achieves -0.73 in TER and +1.49 in BLEU compared to the baseline, and ranks second among all submitted systems.

1 Introduction

Automatic Post-Editing (APE) is the task of automatically correcting errors in a given the machine translation (MT) output to generate a better translation (Chatterjee et al., 2018). Because APE can be regarded as a sequence-to-sequence problem, MT techniques have been previously applied to this task. Subsequently, it is only natural that neural APE has been proposed following the appearance of neural machine translation (NMT).

Among the initial approaches to neural APE, a log-linear combination model (Junczys-Dowmunt and Grundkiewicz, 2016) that combines bilingual and monolingual translations yielded the best results. Since then, In order to leverage information from both MT outputs (*mt*) and its corresponding source sentences (*src*), a multi-encoder model (Libovický et al., 2016) based on multi-source translation (Zoph and Knight, 2016) has become the prevalent approach (Bojar et al., 2017). Recently, with the advent of Transformer (Vaswani et al., 2017), most of the participants in the WMT18 APE shared task proposed Transformer-based multi-encoder APE models (Chatterjee et al., 2018).

Previous multi-encoder APE models employ separate encoders for each input (*src*, *mt*), and combine their outputs in various ways: by 1) sequentially applying attention between the hidden state of the decoder and the two outputs (Junczys-Dowmunt and Grundkiewicz, 2018; Shin and Lee, 2018) or 2) simply concatenating them (Pal et al., 2018; Tebbifakhr et al., 2018). However, these approaches seem to overlook one of the key differences between general multi-source translation and APE. Because the errors *mt* may contain are dependent on the MT system, the encoding process for *mt* should reflect its relationship with the source sentence. Furthermore, we believe that it would be helpful to incorporate information from the source sentence, which should ideally be error-free, in addition to the jointly encoded *mt* in generating post-edited sentence.

From these points of view, we propose a multi-source APE model by extending Transformer to contain a joint multi-source encoder and a decoder that involves a multi-source attention layer to combine the outputs of the encoder. Apart from that, we apply various teacher-forcing ratios at training time to alleviate exposure bias. Finally, we ensemble model variants for our submission. The remainder of the paper is organized as follows: Section 2 describes our model architecture.

* Both authors equally contributed to this work

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 3: Shared Task Papers (Day 2) pages 114–119
Florence, Italy, August 1-2, 2019.

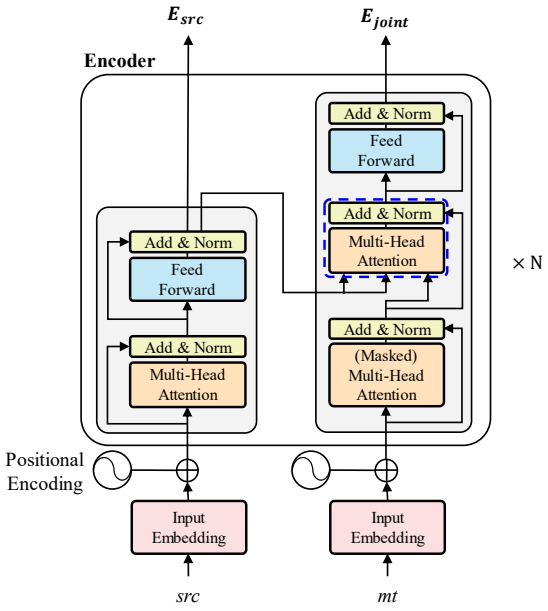


Figure 1: **The architecture of the proposed encoder** – the dashed square indicates the joint hidden representation of two sources

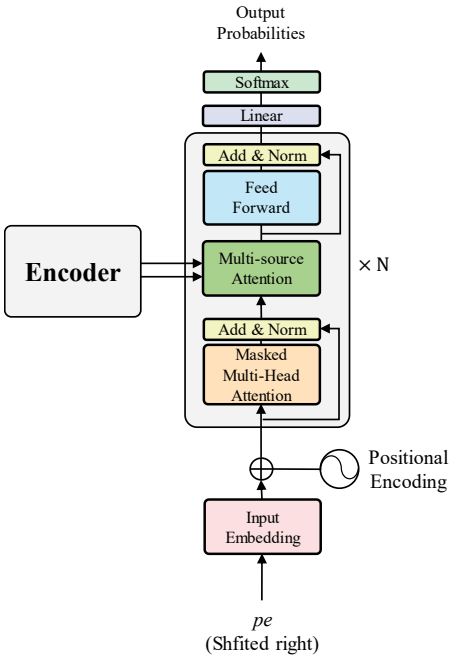


Figure 2: The architecture of the decoder

Section 3 summarizes the experimental results, and Section 4 gives the conclusion.

2 Model Description

We adopt Transformer to the APE problem, which takes multiple inputs (*src*, *mt*) to generate a post-edited sentence (*pe*). In the following subsections, we describe our modified encoder and decoder.

2.1 Encoder

The proposed encoder structure for multi-source inputs, as shown in Figure 1, is an extension of what is introduced in Vaswani et al. (2017) developed considering single-source input. Similar to recent APE studies, our encoder receives two sources: *src* $\mathrm{x} = (x_1, \ldots, x_{T_x})$ and *mt* $\mathrm{y} = (y_1, \ldots, y_{T_y})$, where T_x and T_y denote their sequence lengths respectively, but produce the joint representation $\mathrm{E}_{joint} = (e^j_1, \ldots, e^j_{T_y})$, in addition to encoded *src* $\mathrm{E}_{src} = (e^s_1, \ldots, e^s_{T_x})$.

Joint representation. Unlike previous studies, which independently encode two input sources using separate encoding modules, we incorporate *src* context information into each hidden state of *mt* through the single encoding module, resulting in a joint representation of two sources. As shown with the dashed square in Figure 1, jointly represented hidden states are obtained from the residual connection and multi-head attention that takes $H_{src} \in \mathbb{R}^{T_x \times d}$ as keys and values and $H_{mt} \in \mathbb{R}^{T_y \times d}$ as queries. Therefore, the joint representation of each level of the stack ($i = 1, \ldots, N$) can be expressed with *MultiHead*(Q, K, V) and *LayerNorm* described in Vaswani et al. (2017) as follows:

$$H^i_{joint} = \mathrm{LayerNorm}(H^i_{mt} + C^i_{src})$$

where

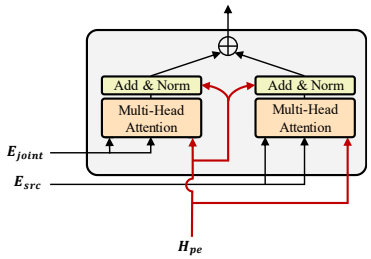
$$C^i_{src} = \mathrm{MultiHead}(H^i_{mt}, H^i_{src}, H^i_{src}) \quad (1)$$

Stack-level attention. When applying attention across source and target, the original Transformer only considers source hidden states retrieved from the final stack, whereas our encoder feeds into each attention layer the *src* embeddings from the same level, as can be seen in (1).

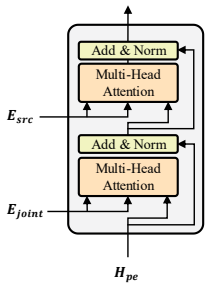
Masking option. The self-attention layer that is the first attention layer of the *mt* encoding module optionally includes a future mask, which mimics the general decoding process of MT systems that depends only on previously generated words. We conduct experiments (§3.2) for two cases: with and without this option.

2.2 Decoder

Our decoder is an extension of Transformer decoder, in which the second multi-head attention layer that originally only refers to single-source



(a)



(b)

Figure 3: Illustrations of the multi-source attention layer. (a) and (b) refer to the linear and sequential combinations, respectively.

encoder states is replaced with a multi-source attention layer. Figure 2 shows our decoder architecture including the multi-source attention layer that attends to both outputs of the encoder. Furthermore, we construct two types of the multi-source attention layer by utilizing different strategies in combining attention over two encoder output states.

Multi-source parallel attention. Figure 3a illustrates the structure of parallel attention. The decoder's hidden state simultaneously attends to each output of the multi-source encoder, followed by residual connection, and the results are linearly combined by summing them at the end:

$$H_{parallel} = H_1 + H_2$$

where

$$H_1 = \text{LayerNorm}(H_{pe} + C_{joint})$$
$$H_2 = \text{LayerNorm}(H_{pe} + C_{src})$$
$$C_{Joint} = \text{MultiHead}(H_{pe}, E_{joint}, E_{joint})$$
$$C_{src} = \text{MultiHead}(H_{pe}, E_{src}, E_{src}).$$

Note that $H_{pe} \in \mathbb{R}^{T_z \times d}$ denotes the hidden states for decoder input *pe* $z = (z_1, \ldots, z_{T_z})$.

Multi-source sequential attention. As shown in Figure 3b, two outputs of the encoder are sequentially combined with the decoder's hidden state: E_{joint} and the decoder's hidden state are first assigned to multi-head attention and residual connection layers, then the same operation is performed between the result and E_{src}.

$$H_{seq} = \text{LayerNorm}(H' + C_{src})$$

where

$$H' = \text{LayerNorm}(H_{pe} + C_{joint})$$
$$C_{src} = \text{MultiHead}(H', E_{src}, E_{src})$$
$$C_{joint} = \text{MultiHead}(H_{pe}, E_{joint}, E_{joint}).$$

This approach is structurally equivalent to Junczys-Dowmunt and Grundkiewicz (2018), except that the encoder states being passed on are different.

Dataset	Triplets	TER
official training set	13,442	14.89
official development set	1,000	15.08
eSCAPE-NMT	7,258,533	60.54
eSCAPE-NMT-filtered	4,303,876	39.65

Table 1: **Dataset statistics** – number of sentence triplets (src, mt, pe) and TER score.

3 Experiments

3.1 Dataset

We used the WMT19 official English-German APE dataset (Chatterjee et al., 2018) which consists of a training and development set. In addition, we adopted the eSCAPE NMT dataset (Negri et al., 2018) as additional training data. We extracted sentence triplets from the eSCAPE-NMT dataset according to the following criteria, to which the official training dataset mostly adheres. Selected triplets have no more than 70 words in each sentence, a TER less than or equal to 75, and a reciprocal length ratio within the monolingual pair (*mt*, *pe*) less than 1.4. Table 1 summarizes the statistic of the datasets.

3.2 Training Details

Settings. We modified the OpenNMT-py (Klein et al., 2017) implementation of Transformer to build our models. Most hyperparameters such as the dimensionality of hidden states, optimizer settings, dropout ratio, etc. were copied from the "base model" described in Vaswani et al. (2017). We adjusted the warm-up learning steps and batch size per triplets to 18k and ~25k, respectively. For data preprocessing, we employed subword encoding (Kudo, 2018) with 32k shared vocabulary.

Teacher-forcing Ratios	Architecture							
	Parallel w/ masking		Parallel w/o masking		Sequential w/ masking		Sequential w/o masking	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
w/o tuning	15.06	77.18	15.03	77.29	14.89	77.38	15.10	77.19
1.00	15.02	77.25	14.95	**77.41**	14.83	**77.54**	14.75	**77.68**
0.95	15.07	77.24	14.94	77.24	14.83	77.41	**14.53**	77.36
0.90	**14.75**	**77.54**	14.94	77.26	14.79	77.40	14.99	77.26
0.85	14.86	77.37	14.95	77.30	**14.73**	77.50	14.76	77.56
0.80	14.98	77.06	**14.93**	77.15	14.83	77.44	15.34	76.79

Table 2: **Results of training variants** – the columns correspond to their architectures and the rows correspond to their teacher-forcing ratios. The bold values indicate the best result in the metrics for each architecture. "w/o tuning" refer to generic model.

Two-step training. We separated the training process into two steps: the first phase for training a generic model, and the second phase to fine-tune the model. For the first phase, we trained the model with a union dataset that is the concatenation of eSCAPE-NMT-filtered, and the upsampled official training set by copying 20 times. After reaching the convergence point in the first phase, we fine-tuned the model by running the second phase using only the official training set.

Model variations. In our experiment, we constructed four types of models in terms of the existence of the encoder future mask and the type of the multi-source attention layer in the decoder as follows:

- **Parallel w/ masking** where the model involves the multi-source parallel attention layer with the encoder mask.
- **Parallel w/o masking** in which the encoder mask is excluded from Parallel w/ masking.
- **Sequential w/ masking** where the model involves the multi-source sequential attention layer with the encoder mask.
- **Sequential w/o masking** in which the encoder mask is excluded from Seq. w/ masking.

Teacher-forcing ratio. During training, because the decoder takes as input the target shifted to the right, the ground-truth words are passed to the decoder. However, at inference time, the decoder consumes only previously produced output words, causing exposure bias. To overcome this problem, we have empirically adjusted the teacher-forcing ratio in the second phase of training, so that teacher-forcing is applied stochastically in such a way that given a ratio α, the greedy decoding output of the previous step is fed into the next input with a probability of $1 - \alpha$.

Ensemble. To leverage all variants in different architectures and teacher-forcing ratios, we combined them using an ensemble approach according to the following three criteria:

- Ens_set_1: top-N candidates among all variants in terms of TER.
- Ens_set_2: top-N candidates for variants in each architecture, in terms of TER.
- Ens_set_3: two candidates for variants in each architecture, achieving the best TER and BLEU scores, respectively.

3.3 Results

We trained a generic model for each of the four model variations mentioned in §3.2. Then, we fine-tuned those models using various teacher-forcing ratios. For evaluation, we used TER (Snover et al., 2006) and BLEU (Papineni et al., 2002) scores on the WMT official development dataset. Table 2 shows the scores of the generic and fine-tuned models according to their architectures and teacher-forcing ratios. The result shows that adjusting teacher-forcing ratio helps improve the post-editing performance of the models.

Table 3 gives the results of the ensemble models. The ensemble models had slightly worse TER scores (+0.02 ~ +0.13) than the best TER score in the fine-tuned variants, but better BLEU scores (+0.09 ~ +0.27) than the best BLEU score. We

Models	TER	BLEU	Submission Name
Ens_set_1-top4	14.66	77.79	–
Ens_set_1-top6	14.62	77.79	–
Ens_set_1-top8	14.62	77.81	–
Ens_set_2-top1	14.58	77.86	Contrastive (top1Ens4)
Ens_set_2-top2	**14.55**	**77.95**	Primary (top2Ens8)
Ens_set_3	14.61	77.86	Contrastive (var2Ens8)

Table 3: **Results of ensemble models** – "Submission Name" indicates the names (types) for the submission. The bold values indicate the best result in each metric.

Systems	TER	BLEU
UNBABEL_Primary	**16.06**	75.96
POSTECH_Primary (top2Ens8)	16.11	**76.22**
POSTECH_Contrastive (var2Ens8)	16.13	76.21
USSAR-DFKI_Contrastive	16.15	75.75
POSTECH_Contrastive (top1Ens4)	16.17	76.15
Tebbifakhr et al. (2018)	16.46	75.53
Junczys-Dowmunt and Grundkiewicz (2018)	16.50	75.44
Shin and Lee (2018)	16.70	75.14
Baseline	16.84	74.73

Table 4: **Submission results** – the top-5 systems among official results of the WMT19 APE shared task. We also include the previous round results for comparison. The bold values indicate the best result in each metric.

selected the three best ensemble models for submission, expecting to reap the benefits from leveraging different architectures in the decoding process. The names and types for submission are noted in Table 3.

Submission results. The results of primary and contrastive submission on the official test set are reported in Table 4. Our primary submission achieves improvements of -0.73 in TER and +1.49 in BLEU compared to the baseline, and shows better results than the state-of-the-art of the last round with -0.35 in TER and +0.69 in BLEU. While our primary system ranks second out of 18 systems submitted this year, it shows the highest BLEU score.

4 Conclusion

In this paper, we present POSTECH's submissions to the WMT19 APE shared task. We propose a new Transformer-based APE model comprising a joint multi-source encoder and a decoder with two types of multi-source attention layers. The proposed encoder generates a joint representation for MT output with optional masking, in addition to the encoded source sentence. The proposed decoder employs two types of multi-source attention layers according to the post-editing strategy. We refine the eSCAPE-NMT dataset and apply two-step training with various teacher-forcing ratios. Finally, our ensemble models showed improvements in terms of both TER and BLEU, and outperform not only the baseline but also the best model from the previous round of the task.

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, and Varvara Logacheva. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, page 169-214.

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: Shared Task Papers*, page 710-725.

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: Volume 2, Shared Task Papers, page 751-758.

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: Shared Task Papers*, page 822-826.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *Proceedings of ACL 2017, System Demonstrations*, 67-72.

Taku Kudo. 2018. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 66-75.

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: Volume 2, Shared Task Papers*, page 646-654.

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)*.

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: Shared Task Papers*, page 827-835.

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*, page 311-318.

Jaehun Shin, and Jong-hyeok Lee. 2018. Multi-encoder Transformer Network for Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, page 840-845.

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*.

Amirhossein Tebbifakhr, Ruchit Agrawal, 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: Shared Task Papers*, page 846-852.

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 *Advances in Neural Information Processing Systems*, page 5998-6008.

Barret Zoph, and Kevin Knight. 2016. Multi-Source Neural Translation. In *Proceedings of NAACL-HLT*, page 30-34.