POSTECH-ETRI's Submission to the WMT2020 APE Shared Task: Automatic Post-Editing with Cross-lingual Language Model

Jihyung Lee[1], **WonKee Lee**[1], **Jaehun Shin**[1]
Baikjin Jung[1], **Young-Kil Kim**[3], **Jong-Hyeok Lee**[1,2]
[1]Departmet of Computer Science and Engineering,
[2]Graduate School of Artificial Intelligence,
Pohang University of Science and Technology (POSTECH), Republic of Korea
[3]Electronics and Telecommunications Research Institute, Republic of Korea
{[1]jihyung.lee, [1]wklee, [1]jaehun.shin, [1]bjjung, [1,2]jhlee}@postech.ac.kr
[3]kimyk@etri.re.kr

Abstract

This paper describes POSTECH-ETRI's submission to WMT2020 for the shared task on automatic post-editing (APE) for 2 language pairs: English–German (En–De) and English–Chinese (En–Zh). We propose APE systems based on a cross-lingual language model, which jointly adopts translation language modeling (TLM) and masked language modeling (MLM) training objectives in the pre-training stage; the APE models then utilize jointly learned language representations between the source language and the target language. In addition, we created 19 million new sythetic triplets as additional training data for our final ensemble model. According to experimental results on the WMT2020 APE development data set, our models showed an improvement over the baseline by TER of -3.58 and a BLEU score of $+5.3$ for the En–De subtask; and TER of -5.29 and a BLEU score of $+7.32$ for the En–Zh subtask.

1 Introduction

Automatic post-editing (APE) is a subtask of MT, which aims to improve MT outputs by directly modifying machine-translated sentences (Chatterjee et al., 2019). Using APE systems to correct such errors that are automatically detectable can greatly reduce human effort compared to correcting machine-translated sentences manually from scratch (Pal et al., 2016).

Given that neural-network systems require a large quantity of training data, creating APE triplets, which each consist of a source sentence (`src`), a machine-translated sentence (`mt`), and a manually post-edited sentence (`pe`), requires a lot of human labor. Furthermore, because neural APE is a recently minted field of study, only a few small-sized training data sets are available at present. To mitigate such data shortage, several methods are proposed such that 1) create artificial APE triplets (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018); and 2) apply 'transfer learning' (Correia and Martins, 2019; Lopes et al., 2019). We believe that pre-trained models such as ELMo (Peters et al., 2018), OpenAI GPT (Radford et al., 2018), and BERT (Devlin et al., 2019) helped APE models learn rich language representations that compensated for the performance loss caused by using an insufficient quantity of training data.

APE is a task that handles both `src` and `mt` simultaneously, and learning a joint representation of these two inputs requires an understanding of both languages. Although previous works that used BERT have shown that transfer learning is effective in APE (Correia and Martins, 2019; Lopes et al., 2019), adopting BERT as a pre-trained language model may restrict to properly model the relation between two different languages because BERT is trained only on monolingual data sets. Therefore, following the recent trend of adopting transfer learning to various NLP tasks, we propose a new method that adopts a cross-lingual language model as a pre-trained langauge model for APE.

2 Related Work

2.1 APE models using BERT-based Encoder-Decoder

Lopes et al. (2019) proposed an APE system to which transfer learning is applied; the system uses multilingual BERT (Devlin et al., 2019) as its pre-trained language model in a Transformer encoder-decoder structure. They also introduced "conservativeness penalty", which discourages the APE system from frequently editing `mt`, into the system. In addition to using BERT as a cross-lingual encoder, they followed Correia and Martins (2019), which used pre-trained BERT to initialize weights of both the encoder and decoder, and shared weights of the

Proceedings of the 5th Conference on Machine Translation (WMT), pages 777–782
Online, November 19–20, 2020. ©2020 Association for Computational Linguistics

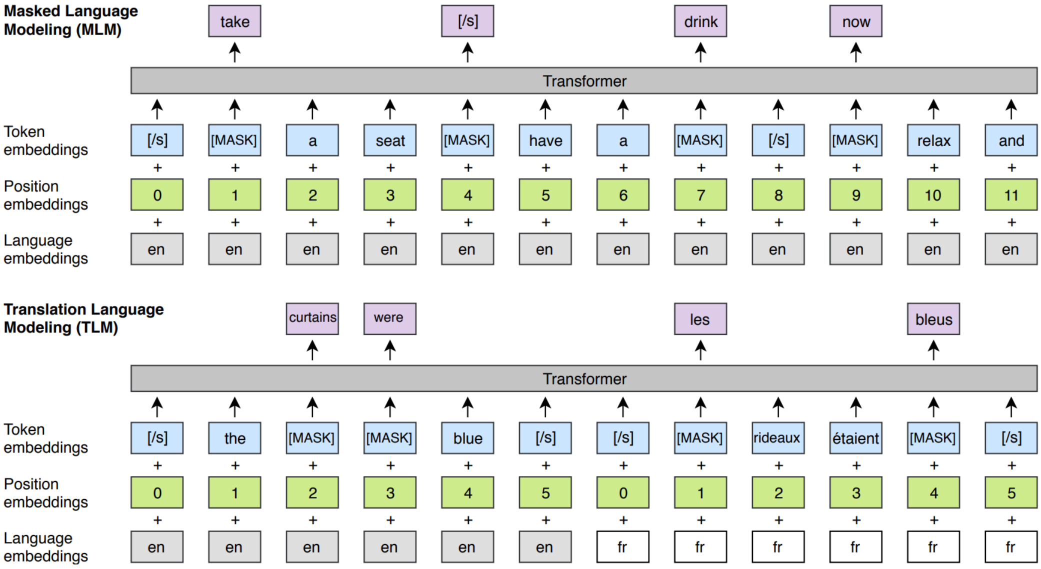
Masked Language Modeling (MLM)
take
[/s]
drink
now
Transformer
Token embeddings
[/s] [MASK] a seat [MASK] have a [MASK] [/s] [MASK] relax and
Position embeddings
0 1 2 3 4 5 6 7 8 9 10 11
Language embeddings
en en en en en en en en en en en en
Translation Language Modeling (TLM)
curtains
were
les
bleus
Transformer
Token embeddings
[/s] the [MASK] [MASK] blue [/s] [/s] [MASK] rideaux étaient [MASK] [/s]
Position embeddings
0 1 2 3 4 5 0 1 2 3 4 5
Language embeddings
en en en en en en fr fr fr fr fr fr

Figure 1: A comparison of the MLM objective and the TLM objective, taken from Conneau and Lample (2019)

self-attention layers both in the encoder and in the decoder.

Furthermore, they used a single encoder that accepts the concatenation of `src` and `mt` as input. To distinguish between the languages, they assigned different segment-embeddings for each language. This BERT-based encoder-decoder system showed the best performance for the English–German (En–De) language pair among all submissions for the WMT2019 APE shared task, proving the effectiveness of transfer learning.

2.2 XLM

After BERT had proposed masked language modeling (MLM), which requires monolingual data only (Devlin et al., 2019), Conneau and Lample (2019) introduced translation language modeling (TLM), which is an extension of MLM and allows the model to use parallel corpora as its input in the pre-training stage; the model can mask any token regardless of its language, and constructs its embedding by considering both sides of the context (Figure 1). The model learns through this process a cross-lingual representation during the training phase.

Considering that the APE task is a cross-lingual task, we expect that learning a cross-lingual representation of two different languages at the pre-training stage will be effective also in APE. Thus, we built a cross-lingual language model, which directly learns the joint representation of the two languages while being trained for the TLM objective, and we supplied it to our system. We describe our proposed model's architecture in the next section.

3 Model Description

Our APE system is built on top of Transformer's encoder-decoder structure (Vaswani et al., 2017). In the following subsections, we describe the main features of the encoder and decoder, respectively. Figure 2 illustrates the overall structure of our model.

3.1 Encoder

Transfer Learning. We built a cross-lingual language model and adopted this pre-trained language model to the encoder. It contains bidirectional and cross-lingual representations of the source and target languages, which are learned from predicting masked tokens from a big quantity of parallel data. Although a MLM+TLM model that was trained in 15 languages has been already released on the XLM GitHub page[1], to use a model that is trained with specific language pair corresponding to `src` and `mt` only, we built new MLM+TLM models. For En–De, we trained our model with the TLM objective on the pre-trained En–De MLM model which is released on the XLM Github page. For En–Zh, we trained our model with both the MLM and TLM objectives from scratch.

[1] https://github.com/facebookresearch/XLM

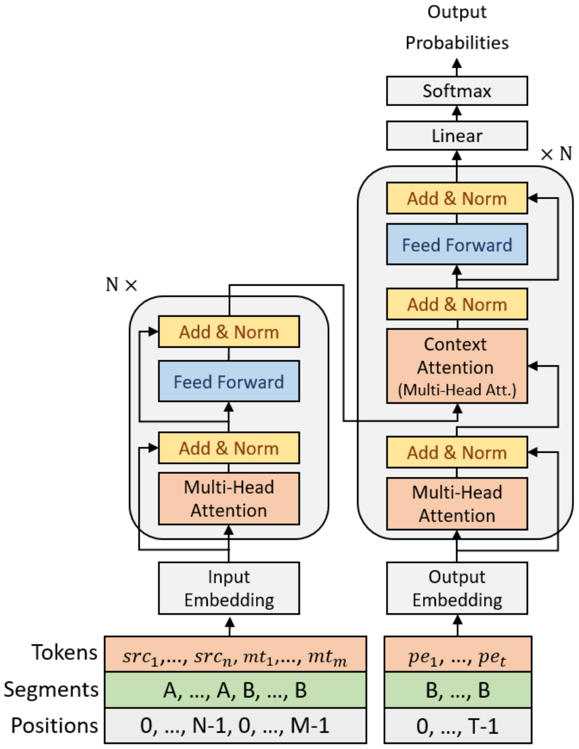

Figure 2: The architecture of our proposed APE model

Input representation. Unlike APE models that use a multi-source encoder that encodes `src` and `mt` separately (Junczys-Dowmunt and Grundkiewicz, 2018; Lee et al., 2019), we followed Lopes et al. (2019) so that the concatenation of `src` and `mt` was fed in to a single encoder. To distinguish one language from the other, we assigned different segment-embeddings to `src` and `mt`, respectively, and we also assigned individual positional-embeddings to `src` and `mt`.

3.2 Decoder

Because our pre-trained language model does not have a decoder, between two options, either randomly initializing the decoder or using another set of pre-trained weights, we chose the former; in contrast to Correia and Martins (2019), who made the encoder's self-attention weights be shared with the decoder, we randomly initialized the context attention layers and did not make the encoder and decoder share their parameters. To compensate for resulting variations in performance, we made an ensemble model of three to four individual models that have identical structures.

Reference	Corpus	En–De	En–Zh
WMT2020 News Translation Task	Europarl v10	✓	–
	ParaCrawl v5.1	✓	–
	Tilde RAPID	✓	–
	Tilde EESC	✓	–
	News Commentary v15	✓	✓
	WikiMatrix	✓	✓
	UN Parallel Corpus	–	✓
	Back-translated news	–	✓
OPUS	Wikipedia	✓	–
	MultiUN	✓	✓
	QED	✓	✓
WMT2019 QE Task Parallel Corpus		✓	–

Table 1: The list of data sets we used to train TLM in the pre-training stage for the En–De & En–Zh language pairs. All data sets were filtered to contain only such sentences with a length between 3 and 70 tokens.

4 Experiments

4.1 Dataset

We applied Byte-Pair Encoding (Sennrich et al., 2016) to all the corpora in both the source and target language. We used the En–De shared sub-word vocabulary that is released on XLM GitHub, but we compiled an En–Zh shared vocabulary by using Wikipedia's dump files in English and Chinese. As in the WMT2020 official data, all English and German data sets were truncated and tokenized with Moses (Koehn et al., 2007) scripts, and the Chinese data set was tokenized with the Jieba tokenizer.[2]

4.1.1 Pre-training stage

We collected parallel corpora from the WMT2020 News Translation Task website,[3] OPUS,[4] and the WMT2019 Quality Estimation website.[5] Table 1 shows the list of parallel corpora that we used to pre-train our models for the two language pairs. To build a pre-trained language model for En–Zh, we built a MLM+TLM model from scratch because we did not have available MLM models that are trained only on the English and Chinese data. Whereas we trained TLM on the whole parallel corpora, we trained MLM only using each side of the parallel corpora as monolingual data. For En–De, we trained only TLM; we used the En-De pre-trained MLM model that is released on the XLM Github page. The sizes of the final parallel corpora that we used in the pre-training stage are 51.7M triplets for En–De and 43.8M for En-Zh.

[2] https://github.com/fxsjy/jieba
[3] http://www.statmt.org/wmt20/translation-task.html
[4] http://opus.nlpl.eu/
[5] http://www.statmt.org/wmt19/qe-task.html

	English-German				English-Chinese			
	WMT20 Dev		WMT20 Test		WMT20 Dev		WMT20 Test	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
Baseline	31.36	50.37	31.56	50.21	60.41	22.62	59.49	23.12
Single	28.39	54.81	–	–	56.25	28.68	–	–
Primary - Top3Ens	**27.78**	**55.67**	27.37	55.83	**55.12**	**29.94**	**54.92**	28.90
Contrastive - Top4Ens	27.94	55.61	**27.02**	**56.37**	55.74	29.69	55.08	**28.97**

Table 2: TER and BLEU scores for En–De and En–Zh language pairs. APE results for the WMT2020 test data will be provided by the shared task organizers. 'Single' is the model which showed the best performance among all the models that later became constituents of the ensemble model.

4.1.2 APE training stage

We used the WMT2018 and WMT2020 official APE data sets for En–De, and the WMT2020 official APE data sets for En–Zh. As supplementary training data, we created new synthetic triplets by following the method to make the eSCAPE NMT data set (Negri et al., 2018); we used the parallel corpora that are released as additional resources for the WMT2020 Quality Estimation task.[6] To create those triplets, we first reused each side of the parallel corpora as `src` and `pe`. We then applied the QE NMT model (Fomicheva et al., 2020)[7] to `src` and then used the resulting translations as `mt`. As a result, we obtained 19M new synthetic triplets for both En–De and En–Zh.

4.2 Training Details

We modified Facebook's XLM implementation that is released on Github[8] to adapt it for the APE task. Most hyperparameters such as the number of layers, the hidden size, and the number of attention heads, were set to those that XLM used for the MT task (Conneau and Lample, 2019). We then used different optimizer-settings for the pre-training and APE training stage, respectively. We used the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 5×10^{-5} in the pre-training stage, and 1×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-6}$ in the APE training. We used 30k warm-up steps and a batch size of 32.

Similar to Lee et al. (2019), we divided the APE training process into two parts. The first is to train the model with 19M triplets, consisting of the 15-times up-sampled WMT official training data and our new synthetic data; we also added the WMT2018 official training data without upsampling only for En-De. This first part took about three days on a single Tesla V100 GPU. The second is to fine-tune the model using only 7k triplets, which are official WMT2020 APE data. This second part took about three hours on the same GPU.

In the decoding stage, we used beam decoding with a beam size of five. We randomly initialized the weights of decoder's context attention layers and experimented our models four times to form an ensemble model of those four models. Our primary model is an ensemble model of three models, excluding one model that scored worst in terms of TER; this model showed the best performance on the WMT2020 development data set. Our contrastive model (Top4Ens) is an ensemble model of all the four models.

4.3 Results

We evaluated our results by comparing them to the MT baseline, which is uncorrected outputs of MT system. We used two evaluation methods that the WMT2020 APE task organizers suggested: Translation Error Rate (TER) and Bilingual Evaluation Understudy (BLEU). We used tercom software[9] to measure TER and a script of XLM GitHub to measure BLEU.

Table 2 describes the results of our proposed model on the WMT2020 official development and test data sets. For the development data set, our 'single' model outperformed the MT baseline in both language pairs. This result implies that our model successfully enhances the original quality of `mt`. Moreover, our primary ensemble model (Top3Ens) showed improvements over the MT baseline: for En–De by TER of -3.58 and by a BLEU score of

[6] http://www.statmt.org/wmt20/quality-estimation-task.html

[7] https://github.com/facebookresearch/mlqe/tree/master/nmt_models

[8] https://github.com/facebookresearch/XLM

[9] http://www.cs.umd.edu/ snover/tercom/

$+5.3$ and for En–Zh by TER of -5.29 and a BLEU score of $+7.32$.

Especially, for the test data set, our contrastive ensemble model showed a significant improvement for En–De by TER of -4.54 and a BLEU score of $+6.16$. For En–Zh, our primary submission showed an improvement over the MT baseline by a big margin: TER of -4.57 and a BLEU score of $+5.78$.

Although we submitted Top3Ens as our primary model, Top4Ens showed better TER and BLEU scores on the En–De test data set. We speculate that this result may have been caused by generality problem in which certain differences between the WMT2020 development and test data sets could occur.

5 Conclusion

For the WMT2020 APE shared task, we propose APE systems that adopt cross-lingual pre-trained language models. To better apply transfer learning to the APE task, we trained TLM in addition to using the original MLM models and initialized the decoder's weights in the same way as the encoder. Furthermore, we created new synthetic triplets to augment the training data and used the ensemble technique to build our final model.

Experimental results show that our proposed model achieved significant improvements on the WMT2020 development and test data sets in terms of TER and BLEU scores for both En–De and En–Zh language pairs.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MISP) (No. 2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH) and R7119-16-1001, Core technology development of the real-time simultaneous speech translation based on knowledge enhancement), and was results of a study on the "HPC Support" Project, supported by the 'Ministry of Science and ICT' and NIPA.

References

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 11–28, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Gonçalo M. Correia and André F. T. Martins. 2019. A simple and effective approach to automatic post-editing with transfer learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3050–3056, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation.

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*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. MS-UEdin submission to the WMT2018 APE shared task: Dual-source transformer for automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 822–826, Belgium, Brussels. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

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 Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

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 112–117, Florence, Italy. Association for Computational Linguistics.

António V Lopes, M Amin Farajian, Gonçalo M Correia, Jonay Trénous, and André FT 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 118–123.

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, Sudip Kumar Naskar, and Josef van Genabith. 2016. 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. The COLING 2016 Organizing Committee.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. 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 *Advances in neural information processing systems*, pages 5998–6008.