HW-TSC's Participation in the WMT 2023 Automatic Post Editing Shared Task

Jiawei Yu[1]*, **Min Zhang**[2], **Yanqing Zhao**[2], **Xiaofeng Zhao**[2], **Yuang Li**[2],
Chang Su[2], **Yinglu Li**[2], **Miaomiao Ma**[2], **Shimin Tao**[2], **Hao Yang**[2]
[1]School of Informatics, Xiamen University, China
[2]Huawei Translation Services Center, Beijing, China
yujiawei@stu.xmu.edu.cn
{zhangmin186,zhaoyanqing,zhaoxiaofeng14,liyuang3,suchang8,liyinglu,
mamiaomiao, taoshimin, yanghao30}@huawei.com

Abstract

The paper presents the submission by HW-TSC in the WMT 2023 Automatic Post Editing (APE) shared task for the English-Marathi (En-Mr) language pair. Our method encompasses several key steps. First, we pre-train an APE model by utilizing synthetic APE data provided by the official task organizers. Then, we fine-tune the model by employing real APE data. For data augmentation, we incorporate candidate translations obtained from an external Machine Translation (MT) system. Furthermore, we integrate the En-Mr parallel corpus from the FLORES-200 dataset into our training data. To address the overfitting issue, we employ R-Drop during the training phase. Given that APE systems tend to exhibit a tendency of 'over-correction', we employ a sentence-level Quality Estimation (QE) system to select the final output, deciding between the original translation and the corresponding output generated by the APE model. Our experiments demonstrate that pre-trained APE models are effective when being fine-tuned with the APE corpus of a limited size, and the performance can be further improved with external MT augmentation. Our approach improves the TER and BLEU scores on the development set by -2.42 and +3.76 points, respectively.

1 Introduction

Automatic Post-Editing (APE) is a post-processing task in a Machine Translation (MT) workflow, aiming to automatically identify and correct errors in MT outputs (Chatterjee et al., 2020a). WMT has been holding APE task competitions in different languages and fields since 2015. Similar to WMT 2022, WMT 2023's APE task still focuses on the En-Mr language pair. Participants are provided with a training set comprising 18,000 instances, a development set, and a test set, with each containing 1,000 instances. Each dataset consists of triplets — the source (*src*) sentences, the corresponding machine-translation (*mt*) outputs, and the human post-edited versions (*pe*) of the translations. In this task, the source sentences have been translated into the target language by using a state-of-the-art neural MT system to get the machine-translation data. The provided data encompasses diverse domains, such as healthcare, tourism, and general/news. In addition, the synthetic training data is offered to participants, which is created by taking a parallel corpus, where the source data is translated using an MT system, and the references are considered as post-edits. Furthermore, participants are permitted to utilize any additional data for systems training.

Typically, training an APE model requires large amount of training data. However, obtaining *pe* is an expensive task in terms of time and money. As a result, there exists a scarcity of large-scale APE datasets.

To address this challenge, numerous data augmentation techniques have been proposed (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018; Lee et al., 2020; Wei et al., 2020; Zhang et al., 2023). Wei et al. (2020) augment the APE training data with translations generated using a different MT system. Huang et al. (2022) train an external MT to obtain more datasets consistent with APE tasks. They also use Google translation to back translate the post-edits in the training set. Deoghare and Bhattacharyya (2022) augment the APE data by generating phrase-level APE triplets using SMT phrase tables. To ensure the quality of the synthetic data, they employe the LaBSE technique (Feng et al., 2022) to filter low-quality triplets.

In our method, we use Google translation to back translate the post-edits in the training set. Subsequently, our dataset is structured as follows: the concatenation of source sentence, back translation and machine translation as the input, while the

[*] Work done during internship at Huawei

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 926–930
December 6–7, 2023. ©2023 Association for Computational Linguistics

post-edits serve as the reference output. Additionally, we incorporate En-Mr parallel sentences from FLORES-200 (Costa-jussà et al., 2022) dev and test data to our training set. Given that we have an En-Mr parallel corpus only and lack machine translation data, we directly utilize English sentences as the source input and Marathi sentences as the post-edits. Furthermore, we use R-Drop (Liang et al., 2021), which regularizes the training inconsistency induced by dropout and has been proven beneficial for different kinds of models.

Chatterjee et al. (2020b) have proven that APE systems often make unnecessary edits to translation output. To mitigate this issue of over-correction, we employ a sentence-level QE system to determine the final output, selecting between the APE system's output and the original machine-translated (*mt*) version.

When being evaluated on the development set, our approach improves the TER (Snover et al., 2006) by -2.42 points and the BLEU score (Papineni et al., 2002) by +3.76 points.

The contributions of our work are as follows:

- We employ two approaches for data augmentation: (1) We utilize Google translation to back translate the post-edits to get *src'*. (2) We add English and Marathi data from the FLORES-200 dataset to our training set.

- We utilize R-Drop to address over-fitting concerns and enhance the generalization capabilities of our model.

- We employ a sentence-level QE system to select the most appropriate output, choosing between the APE-generated output and the original translation.

2 Related Work

Last year's WMT22 APE shared task mainly focuses on transfer learning and data augmentation. Huang et al. (2022) employ the existing data to train an En-Mr translation model as a data augmentation method. Additionally, they utilize an external MT system to generate back-translations, which can be used to add a set of parallel corpora for the model to learn the rules of post-edits. Adapters are also incorporated into the APE model, allowing the training data to be steered to different adapters based on the output of a trained classifier. This facilitates the model in learning post-editing rules specific to different translations.

Deoghare and Bhattacharyya (2022) use two separate encoders to generate representations for src and mt. They also employ a pre-trained language model to initialize the weights for both our encoders. For data augmentation, they leverage external MT candidates and generate phrase-level APE triplets using SMT phrase tables. Furthermore, they filter low-quality APE triplets from the synthetic data using LaBSE-based filtering. They also use a sentence-level QE system to select the final output between the APE-generated output and the original translation.

With experience in previous competitions, we also utilize an external MT system to generate back-translations. Additionally, we adopt a sentence-level QE system for selecting the final output.

3 Dataset

3.1 Data source

We use the WMT22 official En-Mr APE dataset, which consists of a training set and a development set. The training set consists of 18,000 APE triplets across domains, such as healthcare, tourism, and general/news. We first use synthetic data with 2.57M instances to pre-train our model, which was prepared as a part of the 2022 APE shared task. Furthermore, we enrich our training set by incorporating 2,000 En-Mr parallel sentences from the FLORES-200 dataset. FLORES-200 is a high quality, many-to-many benchmark dataset, which contains about 204 languages. In our approach, we specifically extract the English and Marathi parallel corpus from this dataset for training purposes.

4 Model

Figure 1 shows the architecture of our APE model. In this section, we provide the details of our approach.

4.1 Fine-tuned Transformer

We basically treat the APE task as an NMT-like problem, which takes *src* and *mt* as input and generates *pe* autoregressively. Following previous works, we use a special token <*s*> to concatenate src and mt to generate the input sentence: [*src*, <*s*>, *mt*], while the target sentence is *pe*. Initially, we pre-train the APE model using the standard Transformer (Vaswani et al., 2017) structure on 2.57M synthetic training data. However, since there is a mismatch between the synthetic data and the real data in our task, we further fine-tune the APE

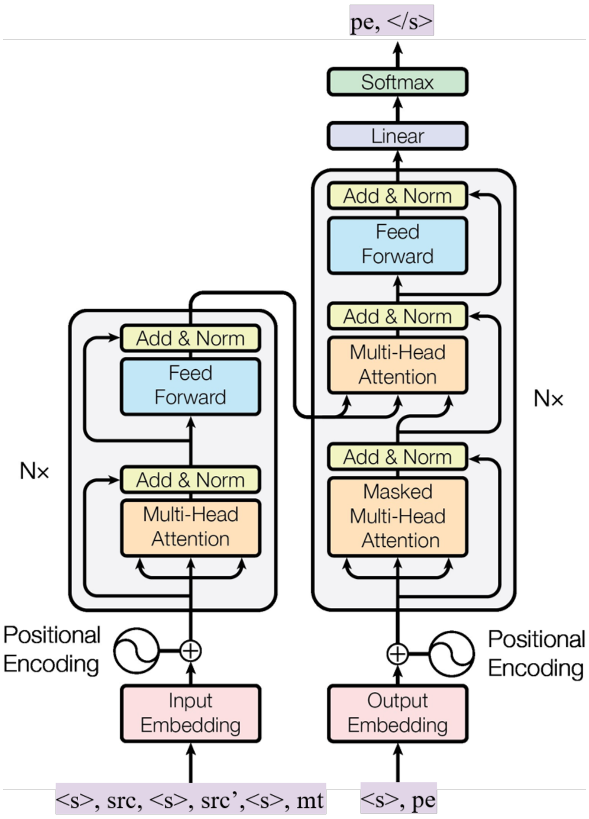

Figure 1: This figure shows the architecture of our model, where mt and augmented src' are concatenated with src before being input into the encoder, and post-edits are generated with the decoder.

model using the APE dataset. To further solve the problem of data scarcity, following (Huang et al., 2022), we use the Google translation system to create the *src'* from the provided *pe* text. We simply concatenate the *src'* with the original *src* and *mt* to form the new input: [*src, <s>, src', <s>, mt*]. Then, we use it in the same way as before, aiming to have the model learn complementary information from *src* and *src'*. During inference, the same input [*src, <s>, src', <s>, mt*] is employed to generate the output, thereby enabling the utilization of the external information derived from *src'*.

We also employ R-Drop during the fine-tuning stage to mitigate overfitting and enhance the generalization capabilities of our model.

4.2 Sentence-Level Quality Estimation

We use wmt22-cometkiwi-da (Rei et al., 2022) as our sentence-level QE model, which is a COMET quality estimation model. This model can be used for reference-free MT evaluation. It receives a source sentence and the respective translation and returns a single score between 0 and 1 that reflects the quality of the translation, where 1 represents a perfect translation. We use this model to rate both the original machine translation and the output generated by our APE system. We then compare the ratings for both sequences and select the one with a higher rating as the final output.

5 Experiment

5.1 Settings

Our model is implemented with fairseq (Ott et al., 2019). Note that the vocabulary and encoder/decoder embeddings of our model are shared between two languages and contain 30K subtokens. All models are trained on a Nvidia Tesla V100 GPU with 32GB memory. We use the batch size of 30,720 tokens in the pre-training stage and 8,192 tokens in the fine-tuning stage. We leverage the FP16 (mixed precision) training technique to accelerate the training process. In all stages, we apply the Adam optimizer(Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ to train the model, where the inverse square root schedule algorithm and warmup strategy are adopted for the learning rate. Concretely, We use a learning rate of 5e-4 with 20k warm-up steps in the pre-training stage and a learning rate of 5e-5 with 4k warm-up steps in the fine-tuning stage. Besides, we set the dropout to 0.1 in the pre-training stage, 0.3 in the fine-tuning stage, and the value of label smoothing to 0.1 in all stages. Early stopping is adopted with patience 10 and 30 epochs during pre-training and fine-tuning, respectively. During inference, we use beam search with a beam size of 10. Finally, we employ BLEU to evaluate the model performance. TER and newly added evaluation metric chrF (Popovic, 2015) are also used to evaluate the model output.

System	BLEU↑	TER↓
Baseline (Do nothing)	64.62	22.93
+APE Data Fine-tuning	66.20	22.82
+External MT	66.46	22.12
+Flores data	66.83	22.01
+R-Drop	67.76	21.12
+Sentence-level QE	**68.38**	**20.51**

Table 1: Results on the WMT23 APE development set. A situation with a higher BLEU score but lower TER indicates a better result.

5.2 Result

Table 1 shows the experimental results evaluated on the dev set, where the baseline result is produced by directly calculating scores between the provided MT and PE.

The first experiment is performed by fine-tuning all parameters of the pre-trained Transformer on the official training set, which obtains 2+ performance gains compared with the baseline. This demonstrates that fine-tuning the pre-trained NMT model on the limited dataset can be useful. The experiment of adding external MT for data augmentation shows significant improvements in performance. The third row in Table 1 shows the results of the experiment where we add FLORES-200 data. In the fourth row, we show the results when R-Drop is adopted in our training stage. Toward the end, we utilize a sentence-level QE system to rate both the original translation and the APE output. We then select one of them with a higher rating as the final output of our APE system. With the combination of the APE model and sentence-level QE system, we see that the TER decreases to 20.50, and the BLEU score increases to 68.38 points.

6 Conclusion

This paper presents our APE system submitted to the WMT 2023 APE English-Marathi shared task. In our approach, we initially employ the data augmentation method to build the [*src, <s>, src', <s>, mt*] additional training datasets. We augment our training data by incorporating the En-Mr parallel sentences from Flores-200 dataset. We mitigate overfitting by employing R-Drop during the training phase. Moreover, we explore the sentence-level QE system to discard low-quality APE outputs. Evaluation of our APE system shows that our approach achieves significant gains on the WMT-22 APE development sets.

Limitations

One limitation of our approach is that while we utilize a sentence-level QE system to assess the quality of the APE output and the original translation, the APE system itself does not directly benefit from this evaluation process. While the QE system helps us identify and discard poor-quality APE outputs, it does not contribute to the improvement of the APE system itself.

Acknowledgements

We would like to thank the anonymous reviewers. Their insightful comments helped us in improving the current version of the paper.

References

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020a. Findings of the WMT 2020 shared task on automatic post-editing. In *Proc. of WMT@EMNLP*.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020b. Findings of the WMT 2020 shared task on automatic post-editing. In *Proc. of WMT@EMNLP*.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*.

Sourabh Dattatray Deoghare and Pushpak Bhattacharyya. 2022. IIT bombay's WMT22 automatic post-editing shared task submission. In *Proc. of WMT*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proc. of ACL*.

Xiaoying Huang, Xingrui Lou, Fan Zhang, and Tu Mei. 2022. Lul's WMT22 automatic post-editing shared task submission. In *Proc. of WMT*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proc. of WMT*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

WonKee Lee, Jaehun Shin, Baikjin Jung, Jihyung Lee, and Jong-Hyeok Lee. 2020. Noising scheme for data augmentation in automatic post-editing. In *Proc. of WMT@EMNLP*.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In *Proc. of NeurIPS*.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. ESCAPE: a large-scale synthetic corpus for automatic post-editing. In *Proc. of LREC*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proc. of NAACL-HLT*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

Maja Popovic. 2015. chrf: character n-gram f-score for automatic MT evaluation. In *Proc. of WMT@EMNLP*.

Ricardo Rei, Marcos V. Treviso, Nuno Miguel Guerreiro, Chrysoula Zerva, Ana C. Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte M. Alves, Luísa Coheur, Alon Lavie, and André F. T. Martins. 2022. Cometkiwi: Ist-unbabel 2022 submission for the quality estimation shared task. In *Proc. of WMT*.

Matthew G. Snover, Bonnie J. Dorr, Richard M. Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA*. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of NeurIPS*.

Daimeng Wei, Hengchao Shang, Zhanglin Wu, Zhengzhe Yu, Liangyou Li, Jiaxin Guo, Minghan Wang, Hao Yang, Lizhi Lei, Ying Qin, and Shiliang Sun. 2020. Hw-tsc's participation in the WMT 2020 news translation shared task. In *Proc. of WMT@EMNLP*.

Min Zhang, Xiaofeng Zhao, Zhao Yanqing, Hao Yang, Xiaosong Qiao, Junhao Zhu, Wenbing Ma, Su Chang, Yilun Liu, Yinglu Li, Minghan Wang, Song Peng, Shimin Tao, and Yanfei Jiang. 2023. Leveraging chatgpt and multilingual knowledge graph for automatic post-editing. In *International Conference on Human-Informed Translation and Interpreting Technology (HiT-IT 2023)*. Accepted for publication.