Dividing Automatic Post Editing into Sub-Tasks: Quality Estimation and Translation Suggestion

Yuanzhan Tang
MA, Computational Linguistics

Brian Porter
PhD, Philosophy

Abstract

Automatic Post Editing (APE) of machine translations has seen significant progress with the application of Transformer-based models. APE models have tended to focus on transfer learning with sequence-to-sequence machine translation models, and on data augmentation to produce synthetic APE data. We investigate an alternative strategy, dividing the APE task into two subtasks: Quality Estimation (QE) and Translation Suggestion (TS). We present a system consisting of two encoder models, trained on Quality Estimation and MLM tasks, and compare its performance to existing APE models.

1 Introduction

While there have been significant improvements in machine translation (MT) systems, human post-editing of machine translations is often still necessary. Automatic Post-Editing (APE) is a machine learning task in which a model automatically corrects the translations output by a machine translation (MT) system, much as a human post-editor would. APE models are trained to correct MT system outputs by learning from human post-editing patterns, both reducing the need for human post-editing and making any necessary human post-editing easier and more efficient. One common benchmark, and a significant source of progress in APE performance in recent years, has been the WMT APE Shared Task.

Participants in the WMT APE Shared Task develop systems to automatically post-edit the outputs of an unknown machine translation system. In 2020 and 2021, WMT provided data for and evaluated submissions on the English-German and English-Chinese language pairs. We focus on the English-Chinese language pair. The WMT English-Chinese data consists of (i) an English source sentence fragment, (ii) a machine translation of that fragment into Chinese, and (iii) a human post-edit improvement of that machine translation.

This sort of data is relatively expensive to produce, and as a result the limited availability of APE data is a significant bottleneck to model performance. Submissions to the WMT APE Shared Task often involve novel data augmentation techniques, such as those presented in Negri et al. (2018), Lee et al. (2020), or Yang et al. (2020).

However, data augmentation can be computationally expensive, as it often involves using a machine translation model to produce large quantities of translations. Many of these submissions also involve transder learning applied to large machine translation models (>500m parameters), which can make it difficult to run these models on modest hardware.[1].

In an attempt to mitigate these issues, we decided to attempt to solve the APE task without the use of significant data augmentation, or the creation of synthetic APE data, using only models that could be trained on a single GPU of relatively modest memory (<12gb).

To do this, we divided the APE task into two sub-tasks: Quality Estimation (QE), and Translation Suggestion (TS). In the QE task, the model attempts to predict the quality of a translation. There is sequence-level QE, in which a model predicts a quality score for an entire sequence, and there is word-level QE, in which a model attempts to predict for each word whether or not that word has been translated correctly. In the TS task, a model generates potential alternative words or sequences for a machine-translated text.

We reasoned that the APE task is, in effect, composed of two parts: identifying which portions of the MT sequence need to be edited, and generating the correct replacements for those portions. These

[1] It is worth noting that both Lee et al. (2020) and Yang et al. (2020), the submissions reported for the WMT2020 APE task, used Tesla V100 GPUs with 32gb of memory per GPU.

are, in effect, the QE and TS tasks. By separating these tasks and assigning each task to a separate model, we hoped to minimize the need for APE-specific training data. We also hoped that this would allow us to separately train two smaller models, rather than having to train one larger model, thus reducing the hardware requirements.

2 Related Work

Previous submissions to the WMT APE Task have largely focused on transfer learning for machine-translation sequence-to-sequence models, such as Sharma et al. (2021), and Oh et al. (2021), Lee et al. (2020) and Yang et al. (2020). These models have achieved state of the art performance on APE.

This impressive APE performance has benefitted from new methods to generate artificial source/machine-translation/human-PE triplets for APE training. One such method, used in Negri et al. (2018) to generate the synthetic corpus eSCAPE, uses publicly available parallel corpora in the source and target languages, machine-translating the source sentence and using the target language parallel as an artificial human post-edit. Other methods based on back-translation have also been used to expand the size of synthetic corpora and to improve performance, as in Lee et al. (2021b).

There have also been novel training and fine-tuning methods developed for APE tasks. Huang et al. (2019) introduced a new "learning to copy" training method for APE models, in which a model effectively learns which portions of a translation should be copied and which should be replaced.

Outside of the WMT APE task, there have also been significant strides in the use of NLP models both to assist human post-editors, and to perform automatic post-editing. For example, Lee et al. (2021a) introduced IntelliCAT, a system designed to assist human post-editors. This is not an automatic post editing system, but it is a system that uses machine learning models to benefit the post editing process. This model uses two Quality Estimation (QE) models, one sequence-level and one word-level. The model also uses one Translation Suggestion (TS) model. Human post editors see the source text, the machine translated text, and the outputs of IntelliCAT's QE and TS models. The use of IntelliCAT improves both the quality and the speed of human post-edits.

3 Experiments

Earlier work on Automatic Post Editing (APE) has focused on sequence-to-sequence learning. Frequently, transfer learning is applied to an encoder-decoder machine translation model, fine-tuning it to produce a post-edited sequence in the target language.

For our experiments, we aimed to investigate whether the APE task could be effectively subdivided into intermediate tasks that could be implemented on simpler architectures. Inspired by IntelliCAT, we divided the APE task into two sub-tasks: predicting which portions of a machine-translated text needed to be replaced, and making the needed replacements. These are essentially the standard Quality Estimation (QE) and Translation Suggestion (TS) tasks, respectively.

To accomplish this, we trained two encoder models; one was trained for Quality Estimation, and one was trained for Translation suggestion. We treated Quality Estimation as a token classification task, and treated Translation Suggestion as a masked language modelling (MLM) task. This allowed both tasks to be completed by an encoder model, without the need for a decoder.

3.1 Baseline models

As in previous work on the APE task, we used a transfer learning approach to compensate for the limited training data available. Unlike previous work, we used a multilingual encoder model as a baseline, rather than using a seq2seq translation model as a baseline.

For our baseline models, we used two instantiations of the XLM-RoBERTa base model (270m parameters) from Conneau et al. (2019). We also tested an mT5 encoder model taken from Xue et al. (2021), but found that the XLM-RoBERTa model performed better. We also concluded that having both models use the same tokenizer streamlined the training and evaluation processes.

3.2 Quality Estimation Model

Our QE model was pretrained on a sample of 2.2 million sequences taken from the UM-Corpus English-Chinese parallel corpus Tian et al. (2014). For this, we used an ELECTRA-style pretraining task. Sequences, consisting of parallel sentence fragments in English and Chinese, were tokenized. Random tokens in both the English and Chinese subsequences were replaced (probability 0.333)

with a randomly chosen token of the same language. The model was trained on a token classification task, identifying which tokens had been replaced. We did not find that increasing the size of the pretraining corpus improved model performance on the downstream QE task.[2]

After this pretraining step, we then trained our QE model on the WMT2021 QE/APE training data. This consists of 7,000 examples, each of which consists of (i) an source English sentence fragment, (ii) the machine translation (MT) of that fragment into Chinese, (iii) a human-posted-edited correction of the machine translation, and (iv) word-labels for each source/MT pair. A Chinese word is labeled "OK" if the human post-editor did not edit or remove the word when producing the post-edited correction; a Chinese word is labeled "BAD" if the human post-editor changed or replaced the word. An English word is labeled "OK" if its translation into Chinese is left untouched by the human post-editor; it is labeled "BAD" if its translation is changed by the human post-editor.

To train our QE model on this data, we used first tokenized the source and MT sequences, and converted the English and Chinese word-labels into token-labels. We then treated the QE task as a token-classification task: the model was trained to predict whether a token was labeled "OK" or "BAD". In other words, the model was trained to predict which tokens of the machine translation would need to be replaced by a post-editor. This is essentially an encoder-only variation on the "learning to copy" training method introduced by Huang et al. (2019), in which they trained an encoder-decoder model to learn which words in the machine translation should be copied to the post-edited correction, and which should be replaced.

3.3 Translation Suggestion Model

Our TS model was trained by fine-tuning XLM-RoBERTa model to predict the masked span of text in the translation. To train the TS model, we used the WMT2020 and WMT2021 benchmark datasets for the English-Chinese APE task, which consisted of 14,000 examples of (i) an English source sequence, (ii), its machine translation into Chinese, and (iii) a human-posted-edited correction of the machine translation. As the purpose of the model is to predict "correct" translation, the pairs of English source sequence and its post-edited translation were concatenated with a separation token, after being tokenized with the shared BPE model.

To train the model, 20% of tokens were randomly selected and replaced with a <mask> token, and the model was trained to predict the masked tokens. To compare performance, we trained two versions of TS models: one with standard sub-word masking, the other with whole word masking, where if one sub-word token of a word is masked, all the other sub-word tokens of the same word will be masked as well.

3.4 Combining the Models

Once the two models were trained on their respective tasks, we evaluated the combined performance. The WMT 2020 test set was fed to the QE model, which predicted the probability that each token would need to be replaced. We set a cutoff on the probabilities, and then masked a token whenever the model's prediction probability fell above the cutoff. We experimented with different cutoffs: 40%, 50%, 60%, and 75%. The outputs are then fed into the TS model, where all masked tokens are replace by the top 1 prediction of the model.

4 Results

We evaluated the results of each component model in our system separately, before evaluating the performance of our overall system.

During training, we evaluated the results of our QE model on the WMT 2021 development set. After training on the UM-Corpus and the WMT2021 QE data, our QE model achieves 94% accuracy on the WMT2021 development set.

During training, we evaluated the results of our TS model on the WMT 2021 and 2020 development set combined. After training on the WMT2020 and WMT2021 APE data, our TS model with whole word masking has a perplexity of 1.33, while the one with sub-word masking has a perplexity of 2.92.

We evaluated the results of our overall system on the WMT2020 test dataset by comparing them to the official WMT baseline. This baseline is the uncorrected outputs of the MT system. Beating this baseline means improving the machine translation

[2]We originally collected two other parallel corpora from Tiedemann (2012) and El-Kishky et al. (2020). However, these additional corpora did not improve model performance. We believe that the UN corpus from Tiedemann (2012) may be too domain-specific to improve performance, while the average translation quality in the web-scraped CCAligned100 corpus from El-Kishky et al. (2020) was too low to improve performance.

System	TER	sacreBLEU
baseline (MT)	59.49	11.44
Subword Labels	57.26	14.92
Wholeword Labels	57.37	14.92
Subword 40	**52.61**	**15.46**
Wholeword 40	52.78	15.46
Subword 50	53.78	15.28
Wholeword 50	54	15.27
Subword 60	56.56	15.16
Wholeword 60	56.71	15.15
Subword 75	60.23	15
Wholeword 75	60.31	15

Table 1: Results on the WMT 2020 APE test set.

System	TER	BLEU
baseline (MT)	59.49	23.12
POSTECH-ETRI_XLM-Top4Ens_CONTRASTIVE	55.08	28.97
POSTECH-ETRI_XLM-Top3Ens_PRIMARY	54.92	28.90
HW-TSC_DIRECT_CONTRASTIVE.pe	48.01	37.32
HW-TSC_CONCAT_PRIMARY.pe	47.36	37.69

Table 2: Results from the WMT 2020 APE Shared Task. These submissions were evaluated on the same test set as our models. Note that this chart uses a different BLEU score than Table 1, and so cross-table comparisons should only be made using TER.

to more closely resemble the human post-edited correction.

We evaluated our system using the two metrics that the WMT organizers use to evaluate APE submissions: Translation Error Rate (TER) and Bilingual Evaluation Understudy (BLEU). We used the sacreBLEU metric proposed by Post (2018), which resulted in different BLEU scores than those presented in the official WMT2020 results in Chatterjee et al. (2020).

See Table 1 for the TER and sacreBLEU scores for each of our systems. Our best-performing system used the TS model trained with subword masking, and used a 40% cutoff on the QE model's predictions. This model achieved a -6.88 TER improvement, and a +4.02 sacreBLEU improvement over the baseline.

Our results suggest that there is very little performance difference between the TS model trained on whole-world masking vs the TS model trained on sub-word masking, although the model trained on sub-word masking does consistently perform *slightly* better.

Table 2 shows the official results of the WMT 2020 APE Shared Task. Based on TER scores, our best model (Subword 40) falls squarely in the middle of the four submissions. We take this to show that our method of dividing the APE task into subtasks is a viable option, resulting in performance that is comparable to fine-tuning a machine translation model directly on APE tasks using synthetic APE data.

5 Discussion

Our system produces better results given a lower cutoff. This is relatively unsurprising: the higher the cutoff used, the fewer chances the TS model has to improve the translation.

But the most surprising aspect of our results is that the TS model performs better with the QE model's outputs at a 40% or 50% cutoff than it does with the ground truth token-labels taken from the WMT's word-labels. This means that the TS model does *worse* with more accurate inputs.

This is a strange result, and one that we cannot confidently diagnose. However, our best hypothesis is that some of the subtler translation errors that the post-editors correct are difficult for the QE model to identify, but also difficult for the TS model to correct. For example: if there's an egregiously wrong word choice, it may be easy for the QE model to identify the error; it is similarly easy for the TS model to provide with the right word. But if there is a more subtle change to the grammar or sentence structure, the QE model may be unlikely to identify the error in the MT output. But this ends up improving the overall output of the model, because the TS model would be unlikely to correct the error anyway.[3]When we use the ground truth labels, the TS model is forced to unmask *every* mistranslated word. In effect, this means that the TS model is forced to make judgments about even the most difficult cases, which the QE model would ordinarily fail to identify. If the TS model performs poorly enough in unmasking tokens in these difficult cases, that would explain the drop in

[3]The fact that our model performs *worse* that the baseline when using a 75% cutoff suggests that the TS model does in fact make some "corrections" that *reduce* translation quality. It's just that when given enough opportunities, the TS model's good corrections will outnumber the bad corrections.

performance when using the WMT's ground truth labels. The overall effect is that the QE model only tells the TS model to make the changes that the TS model is most likely to make correctly, which improves the overall score. When fed the ground truth labels, the TS model's errors in the difficult cases reduce the overall scores.

Our system performs comparably to the submissions to the WMT2020 APE task. Our system outperforms both POSTECH-ETRI submissions by Lee et al. (2020), but underperforms against the two HW-TSC submissions by Yang et al. (2020). All four of these submissions treat the APE task as a sequence-to-sequence task, and use encoder-decoder transformer models as in Vaswani et al. (2017).

The POSTECH-ETRI team created a synthetic APE dataset, inspired by the eSCAPE dataset from Negri et al. (2018). Creating synthetic APE datasets is much more resource-intensive than creating the ELECTRA-style pretraining dataset that we used for our QE model. The fact that our system outperforms the POSTECH-ETRI submissions suggests that the creation of eSCAPE-style synthetic training data may be unnecessary.

The HW-TSC team do not create their own synthetic APE data, but they do perform some basic data augmentation. They used an additional MT systems to produce a second machine translation of the English source sequences, and feed the English source sequence, the original WMT MT translation sequence, and their own MT translation sequence as model inputs during training. However, the HW-TSC team's model outperforms ours even without this data augmentation: their model achieves a TER of 49.257 without augmented data Yang et al. (2020). Their primary architectural innovation is to add bottleneck adaptor layers (BAL) to their model, as proposed by Houlsby et al. (2019). They introduce these layers to solve the problem of overfitting; without the layers, they find that their models begin overfitting within the first few epochs. We found the same problem when training out models; our QE model began to overfit on the WMT training data after only 3 epochs, while our TS model began to overfit after only 2 epochs.

We believe that the use of these BALs would similarly improve our system's performance. It would be an interesting question for future research to determine whether BALs could improve the performance of our "double encoder" system to the same extent that they improved HW-TSC's encoder-decoder system.

References

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 shared task on automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 5960–5969, "Online". Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp.

Xuancheng Huang, Yang Liu, Huanbo Luan, Jingfang Xu, and Maosong Sun. 2019. Learning to copy for automatic post-editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6122–6132, Hong Kong, China. Association for Computational Linguistics.

Dongjun Lee, Junhyeong Ahn, Heesoo Park, and Jaemin Jo. 2021a. IntelliCAT: Intelligent machine translation post-editing with quality estimation and translation suggestion. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 11–19, Online. Association for Computational Linguistics.

Jihyung Lee, WonKee Lee, Jaehun Shin, Baikjin Jung, Young-Kil Kim, and Jong-Hyeok Lee. 2020. POSTECH-ETRI's submission to the WMT2020 APE shared task: Automatic post-editing with cross-lingual language model. In *Proceedings of the Fifth Conference on Machine Translation*, pages 777–782, Online. Association for Computational Linguistics.

WonKee Lee, Baikjin Jung, Jaehun Shin, and Jong-Hyeok Lee. 2021b. Adaptation of back-translation to automatic post-editing for synthetic data generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational*

Linguistics: Main Volume, pages 3685–3691, Online. Association for Computational Linguistics.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. ESCAPE: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Shinhyeok Oh, Sion Jang, Hu Xu, Shounan An, and Insoo Oh. 2021. Netmarble AI center's WMT21 automatic post-editing shared task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 307–314, Online. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Abhishek Sharma, Prabhakar Gupta, and Anil Nelakanti. 2021. Adapting neural machine translation for automatic post-editing. In *Proceedings of the Sixth Conference on Machine Translation*, pages 315–319, Online. Association for Computational Linguistics.

Liang Tian, Derek F. Wong, Lidia S. Chao, Paulo Quaresma, Francisco Oliveira, Yi Lu, Shuo Li, Yiming Wang, and Longyue Wang. 2014. UM-corpus: A large English-Chinese parallel corpus for statistical machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1837–1842, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

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

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Hao Yang, Minghan Wang, Daimeng Wei, Hengchao Shang, Jiaxin Guo, Zongyao Li, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, and Yimeng Chen. 2020. HW-TSC's participation at WMT 2020 automatic post editing shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 797–802, Online. Association for Computational Linguistics.