BJTU-Toshiba's Submission to CCMT 2021 QE and APE Task

Hui Huang[1], Hui Di[2], Hongjie Ren[1], Kazushige Ouchi[2], Jiahua Guo[1], Hanming Wu[1], Jian Liu[1], Yufeng Chen[1], and Jin'an Xu[1(✉)]

[1] School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
{18112023,20125222,17301093,17271137, jianliu,chenyf,jaxu}@bjtu.edu.cn
[2] Research & Development Center, Toshiba (China) Co., Ltd., Beijing, China
dihui@toshiba.com.cn, kazushige.ouchi@toshiba.co.jp

Abstract. This paper presents the systems developed by Beijing Jiaotong University and Toshiba (China) Co., Ltd. for the CCMT 2021 quality estimation (QE) and automatic-post editing (APE) task. For QE task, we mainly rely on multiple pretrained language models, and propose a multi-phase pre-finetuning scheme, to adapt the pretrained models to the target domain and task. The pre-finetuning scheme consists of language-adaptative finetuning, domain-adaptative finetuning and task-adaptative finetuning. For APE task, we use BERT-initialized Transformer as the backbone model, and create different groups of synthetic data by different data augmentation methods, i.e. forward translation, round-trip translation and multi-source denoising autoencoder. Multi-model ensemble is adopted in both tasks. Experiment results on the development set show high accuracy on both QE and APE tasks, demonstrating the effectiveness of our proposed methods.

Keywords: Machine translation · Quality estimation · Automatic post-editing

1 Introduction

This paper presents the systems developed by Beijing Jiaotong University and Toshiba (China) Co., Ltd. for the CCMT 2021 quality estimation (QE) and automatic-post editing (APE) task. For QE, we participate in the sentence-level task of Chinese-English direction. For APE, we participate in the task of Chinese-English direction.

Machine translation quality estimation aims to evaluate the quality of machine translation automatically without golden reference [2]. The quality can be measured with different metrics, such as HTER (Human-targeted Edit Error) [18]. Machine translation automatic post-editing aims to fix recurrent errors

J. Su and R. Sennrich (Eds.): CCMT 2021, CCIS 1464, pp. 25–38, 2021.
https://doi.org/10.1007/978-981-16-7512-6_3

made by a certain decoder given the source sentence, by learning from correction examples [4]. Both the two tasks serve as a post-processing procedure for machine translation (MT) and are inner-related.

Both tasks rely on human-annotated triplets. QE is trained with triplets of *src* (source sentence), *mt* (machine translated sentence) and *score* (human-assessed score), and APE is trained with triplets of *src*, *mt* and *pe* (post-edited sentence). Since both human-assessment and post-editing require professional translators to manually annotate *src*-*mt* pairs, both tasks are highly data-scarce with only 10k-20k training examples. How to train a accurate estimator or post-editor with limited data remains a challenge.

For QE task, our system mainly relies on multiple pretrained models, including four multilingual pretrained models, i.e. multilingual BERT [8], XLM [6], XLM-RoBERTa-base and XLM-RoBERTa-large [5], and one monolingual model, i.e. RoBERTa [16]. We propose a multi-phase pre-finetuning scheme, to adapt the pretrained model to the target domain and task. The pre-finetuning procedure includes language-adaptative finetuning (LAF), domain-adaptative finetuning (DAF) and task-adaptative finetuning (TAF). We also jointly train the sentence-level estimator with word-level QE task. Different models are ensembled to achieve further improvement.

For APE task, we choose BERT-initialized Transformer [7] as the back-bone model, which uses the pretrained BERT to initialize the parameters of both encoder and decoder. We create synthetic triplets from openly-available parallel data using different methods, i.e. forward translation [17], round-trip translation [12] and multi-source denoising autoencoder. We build the multi-source denoising autoencoder to restore the corrupted reference given the source text, and the restored reference is deemed as the synthetic *mt*. We apply domain-selection to the parallel data for creating synthetic data, and different models trained with different data are ensembled to achieve further improvement.

Experiments on the development set shows we obtain competitive results in both directions, verifying the effectiveness of our proposed method.

2 Chinese-English Sentence-Level Quality Estimation

2.1 Model Description

Given the data-scarcity nature of QE, we build our system based on multiple pretrained models. We mainly rely on four multilingual pretrained models, i.e. multilingual BERT (abbreviated as mBERT) [8], XLM [6], XLM-RoBERTa-base (abbreviated as XLM-R-base) and XLM-RoBERTa-large (abbreviated as XLM-R-large) [5]. All of these four models are based on multi-layer Transformer [22] architecture, and are pretrained on massive multilingual text with shared multilingual vocabulary, enabling them to transfer to downstream tasks with limited training data.

We concatenate *src* (source sentence) and *mt* (machine translated sentence) following the way pre-trained models treat sentence pairs, and then feed the sentence pair to the model. We try two different strategies to aggregate the

sentence-level representation, the first one is to directly use the first hidden representation of the pretrained model, and the second one is to add a layer of RNN on the top of the model, to better leverage the global context information, as shown in Fig. 1.

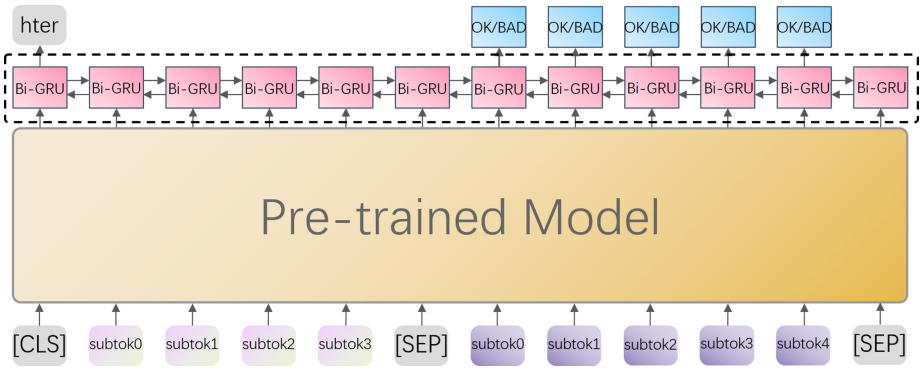

Fig. 1. Pretrained model for quality estimation with joint training. **[CLS]**, **[SEP]** are predefined segment separators, and could be different in different models. The component circled with dashed line is alternative.

Although we mainly focus on sentence-level QE, the sentence and word-level QE are highly related, since their quality annotations are commonly based on the HTER measure [14]. During the calculation of sentence-level HTER score, the word-level QE tag for each word in *mt* could also be derived, and can serve as a supplementary information for training. Therefore, we implement multi-task learning, jointly train the sentence and word-level estimator together. The word-level estimation is based on the output logit according to each word, and we only use the logit of the first sub-token if one word is segmented into multiple sub-tokens. The loss function of both levels are defined as follow:

$$L_{word} = \sum_{s \in D} \sum_{x \in s} -(p_{ok} \log p_{ok} + \lambda p_{bad} \log p_{bad}),$$

$$L_{sent} = \sum_{s \in D} \parallel sigmoid(h(s)) - hter_s \parallel,$$

where s and x denote each sentence and word in the dataset D, and $h(s)$ is the hidden representation, and λ is a hyper parameter. Notice the quality of *mt* is very high [19], which means most of word-level tags are OK. To force the model to pay more attention to the erroneously translated words, we assign a weight λ for BAD words when calculating word-level loss. The loss of both sentence and word level are combined and back-propagated together, defined as follow:

$$L_{joint} = \sum_{s \in D} (L_{sent} + \eta \sum_{x \in s} L_{word}),$$

where η is a coefficient to balance the word-level and sentence-level loss. Since the linear transformation for different levels are implemented on different positions, we can perform multi-task training and inference naturally without any structure adjustment. During the joint-training procedure, the word-level tags can provide fine-grained information for sentence-level QE.

Table 1. Results on the development and test sets of CCMT 2021 Chinese-English sentence-level QE with different pretrained models. We do not apply joint training for XLM-R-large due to time limitation, and the result on dev set for XLM-R-large is very low because we set the max length very short in training.

Model	Method	Dev Set		Test set	
		Pearson	Spearman	Pearson	Spearman
mBERT	w/o joint train	0.5783	0.4768	0.5460	0.4748
	w/ joint train	0.5403↓	0.4339	0.5353↓	0.4254
XLM	w/o joint train	0.5464	0.4627	0.5368	0.4668
	w/ joint train	0.5388↓	0.4647	0.5335↓	0.4601
XLM-R-base	w/o joint train	0.5445	0.5077	0.4887	0.4443
	w/ joint train	0.5371↓	0.5143	0.4816↓	0.4388
XLM-R-large	w/o joint train	0.3643	0.3312	0.4736	0.4510

However, as shown in Table 1, joint training leads to degradation in all directions. This is not consistent with previous works which also apply joint training [11,15]. In the end, we decide to keep all the models for ensemble.

2.2 Multi-phase Pre-finetuning

Fine-tuning pre-trained language models on domain-relevant unlabeled data have become a common strategy to adapt the pretrained parameters to downstream tasks [9]. Previous works also demonstrate the necessity of pre-finetuning when performing QE on pretrained models [10,15]. In our system, we propose a multi-phase pre-finetuning scheme, consisting of language-adaptative finetuning (LAF), domain-adaptative finetuning (DAF), and task-adaptative finetuning (TAF). We pre-finetune the pretrained model on unsupervised parallel data with no quality annotations, by continuing performing mask language modeling.

LAF aims to adapt the pretrained model to bilingual concatenated pairs. Despite the shared multilingual vocabulary and training data, mBERT and XLM-R are originally monolingually trained, treating the input as either being from one language or another. But in our scenario, the input sentence pair is the

Table 2. Results on the development and test sets of CCMT 2021 Chinese-English sentense-leve QE. We do not apply LAF to XLM-R-large due to limited computation resource, and the result on dev set for XLM-R-large is very low because we set the max length very short in training.

Model	Method	Dev set		Test set	
		Pearson	Spearman	Pearson	Spearman
mBERT	Original	0.5783	0.4768	0.5460	0.4748
	+LAF	0.5875	0.4851	0.5547	0.4824
	+DAF	0.5933	0.4924	0.5589	0.4859
	+TAF	**0.5995**	0.5028	**0.5647**	0.4910
XLM	Original	0.5464	0.4627	0.5368	0.4668
	+DAF	0.5915	0.5065	0.5811	0.5053
	+TAF	**0.5942**	0.5304	**0.5838**	0.5077
XLM-R-base	Original	0.5445	0.5077	0.4887	0.4443
	+LAF	0.5699	0.5164	0.5110	0.4555
	+DAF	**0.5754**	0.5170	**0.5159**	0.4599
	+TAF	0.5716	0.5265	0.5103	0.4639
XLM-R-large	Original	**0.3643**	0.3312	0.4736	0.4510
	+DAF	0.3296	0.2996	0.5237	0.4961
	+TAF	0.2941	0.2674	**0.5379**	0.5090

concatenation of a bilingual parallel pair from two different languages. Therefore, we continue the mask language model on massive parallel sentence pairs (Table 2).

We use the parallel data from CCMT 2021 Chinese-English translation task, which contains roughly 9 million sentence pairs. We filter the data according to length and length ratio, and only keep sentence pairs with length shorter than 60, since we are unable to pre-finetune the pretrained model with *max_len* too big. The remaining 6 million pairs are used for LAF, which takes us roughly 10 days on two GPUs.

On the contrary, XLM is pretrained with the task of Translation Language Modeling, therefore we believe it is already adapted to bilingual concatenated sentence pair. Since LAF is performed on massive data with high computation overhead, we decide not to perform LAF on XLM.

DAF aims to adapt the pretrained model to the target domain. The representation of pretrained model is learned from the combination of various domains, and can be adapted to a certain domain if continued finetuning on unlabeled data from the domain. To this end, we select a domain-similar subset of the parallel data, and perform DAF for all the four pretrained models.

To be more specific, we finetune BERT as the domain classifier. The sentence pairs in the training and development set are deemed as in-domain data, and we randomly sample the same size of data as the general-domain data, for the

training of classifier. We keep roughly 100k domain-similar sentence pairs for DAF, which takes us up to 3–4 hours on a single GPU.

TAF refers to pre-finetuning on the unlabeled training set for the given task. It uses a far smaller corpus (10k pairs) compared to DAF, but the data is much more task-relevant. We apply TAF for all the four models, and it is very fast with no more than 1 h on a single GPU.

The three-phase finetuninig scheme is performed in a pipelined manner, namely the latter phase is performed based on the parameters of the former phase. The representation of the pretrained model is adapted to our target language, domain and task, and can serve as a better start point to be finetuned on downstream task. Despite the limited training data, parallel data is readily accessible, therefore multi-phase finetuning is a convenient yet effective method to improve the performance without extra annotation.

2.3 Partial-Input Estimation

As denoted by Sun [20], QE systems trained on partial inputs perform as well as systems trained on the full input. Although the alignment information is absent, estimation can still be performed solely on the source text (to estimate the complexity) or solely on the target text (to estimate the fluency). This enables the incorporation of powerful monolingual models.

In our system, we perform partial-input estimation on the target side. We utilize the monolingual models of BERT and RoBERTa [16] to estimate the fluency. Only the target side of the bilingual pair is fed for training and evaluation. Despite the absence of the source text, the partial-input estimation still achieve high correlation because of the introduction of powerful monolingual model.

We also perform DAF and TAF to the monolingual model to adapt it to our scenario, as shown in Table 3.

Table 3. Results on the development and test sets of CCMT 2021 Chinese-English sentense-leve QE with partial-input.

Model	Method	Dev set		Test set	
		Pearson	Spearman	Pearson	Spearman
BERT-base	Original	0.5127	0.4652	0.4595	0.4177
RoBERTa-base	Original	0.5471	0.4656	0.4707	0.4279
RoBERTa-large	Original	0.5684	0.5133	0.4785	0.4350
	+DAF	**0.5715**	0.5407	**0.4903**	0.4457
	+TAF	0.5712	0.5063	0.4834	0.4395

2.4 Model Ensemble

After exhaustive hyper-parameter searching, we obtain more than ten strong models with different architectures and training procedures. To combine different predictions and achieve further improvement, we try two model ensemble techniques, namely averaging and linear regression. Averaging simply averages the predicted logits of different models. Linear regression learns a linear combination of different predictions using l_2-regularized regression over the dev set.

Table 4. Results on the development and test sets of CCMT 2021 Chinese-English sentense-leve QE. The results of single models are inconsistent with previous sections due to our final hyper-parameter searching.

Model	Dev set	Test set
	Pearson	Pearson
mBERT	0.6125	0.5581
XLM	0.6055	0.5800
XLM-R-base	0.5974	0.5454
XLM-R-large	0.2941	0.5379
RoBERTa	0.5681	0.4903
Averaging	0.6291	**0.6043**
Linear regression	**0.6376**	0.6034

As shown in Table 4, both two ensemble techniques achieve considerable improvement. Although the result of partial-input is comparatively low, it can provide complimentary information for other bilingual models when doing ensemble. Therefore, the incorporation of partial-input estimation is necessary.

3 Chinese-English Automatic Post-Editing

3.1 BERT-initialized Transformer

The current state of the art in APE is based on encoder-decoder structure with Transformer [22] as the backbone network. To alleviate the data-scarcity problem, we follow [7] and use multilingual BERT to initialize the parameters of Transformers, as shown in Fig. 2, which we call BERT-initialized Transformer. We follow their default setting, namely use the self-attention in BERT to initialize both the encoder and the decoder.

Specifically, instead of using multiple encoders to separately encode *src* and *mt*, we use BERT pre-training scheme, where the two strings after being concatenated by the **[SEP]** special symbol are fed to the single encoder, and assign different segment embeddings to each of them. Both the self-attention and context attention of the decoder are initialized with BERT. The self-attention and

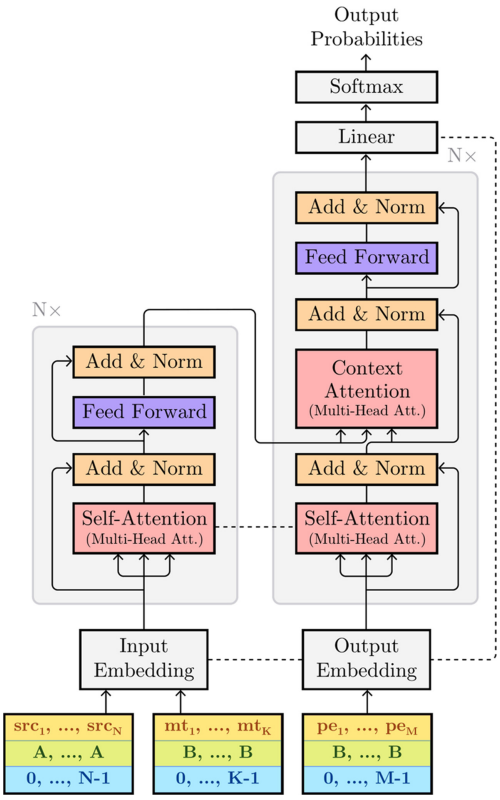

Fig. 2. BERT-initialized Transformer. Dashed lines show shared parameters.

Table 5. Results on the development set of CCMT 2021 Chinese-English APE with different architectures.

Model	Data	Dev set	
		TER	BLEU
Dual-source Transformer [13]	2 million	0.4585	41.74
Multi-source Transformer [21]	2 million	0.4344	46.39
BERT-based Transformer [7]	2 million	**0.4140**	**46.58**

embedding between encoder and decoder are shared, to reduce parameter size and improve training efficiency (Table 5).

We also compare with the dual-source transformer architecture of [13] and multi-source Transformer architecture of [21]. With 10k training triplets combined with 2 million synthetic triplets, the BERT-based Transformer outperforms the previous methods by a large margin, showing the effectiveness of pre-trained parameters in APE task.

3.2 Domain Selection

Firstly we believe generative task is data-hungry, and therefore we use all the available parallel data to create synthetic triplets. We use the parallel data provided by CCMT 2021 Chinese-English translation, which consists of 23 million sentence pairs after filtering. However, during training we find that the model converges very soon and can not be improved afterwards. Therefore, we decide to apply domain selection for the synthetic data.

Table 6. Results on the development set of CCMT 2021 Chinese-English APE with different size of synthetic data. 10k refers to the model trained only with real data.

Model	Data	Dev Set	
		TER	BLEU
BERT-based Transformer	10k	0.4234	45.92
BERT-based Transformer	23 million	0.4679	39.57
BERT-based Transformer	5 million	0.4276	44.62
BERT-based Transformer	1 million	0.4089	47.80
BERT-based Transformer	200k	**0.4011**	**48.86**

To perform domain classification, we use the 10k training triplets as in-domain data, and randomly sample the same size of general domain data. We try two domain classification methods, [1] finetune BERT as a binary classifier, [2] use bilingual cross-entropy filtering method [1], and we use kenlm[1] to train 4-gram language models for filtering. Then synthetic triplets are combined with real triplets (which is oversampled 20 times) for training.

However, we do not see a clear difference between the two domain selection methods. On the contrary, we find that data size matters a lot. As shown in Table 6, we get the best result when incorporating 200k data. More data leads to domain irrelevance while only using the 10k real data is not enough for training. Therefore, we adopt the same data size in the following experiments.

3.3 Data Augmentation Techniques

Data augmentation is a de-factor paradigm for APE task [3]. The creation of synthetic data requires to generate synthetic *mt* given the parallel data (which are deemed as synthetic *src* and *pe*). Previous works rely on translation model to generate synthetic *mt* [12,17], but the connection between synthetic *mt-pe* is not consistent with real *mt-pe*. Actually, most synthetic *mt*s generated by machine translation are a correct translation of *src* but with different syntactic structure from *pe*. Forcing the APE model to transform the syntax of a correct translation is of little help to the training objective.

[1] https://kheafield.com/code/kenlm/.

In this work, we propose to generate synthetic *mt* via Multi-source Denoising Autoencoder (MDA), to better simulate the real error distribution. Denoising autoencoder is trained with two steps: (1) corrupt the text with an arbitrary noising function, and (2) learn a sequence-to-sequence model to reconstruct the original text. Specifically, in our scenario, we provide both the corrupted text and its corresponding translation to the encoder, leading to a multi-source denoising autoencoder structure, as shown in Fig. 3. The MDA learns to reconstruct the text based on its corruption and corresponding translation. This procedure is performed on massive publicly-available parallel sentence pairs (which are denoted as *src* and *ref*), without the need of extra annotations.

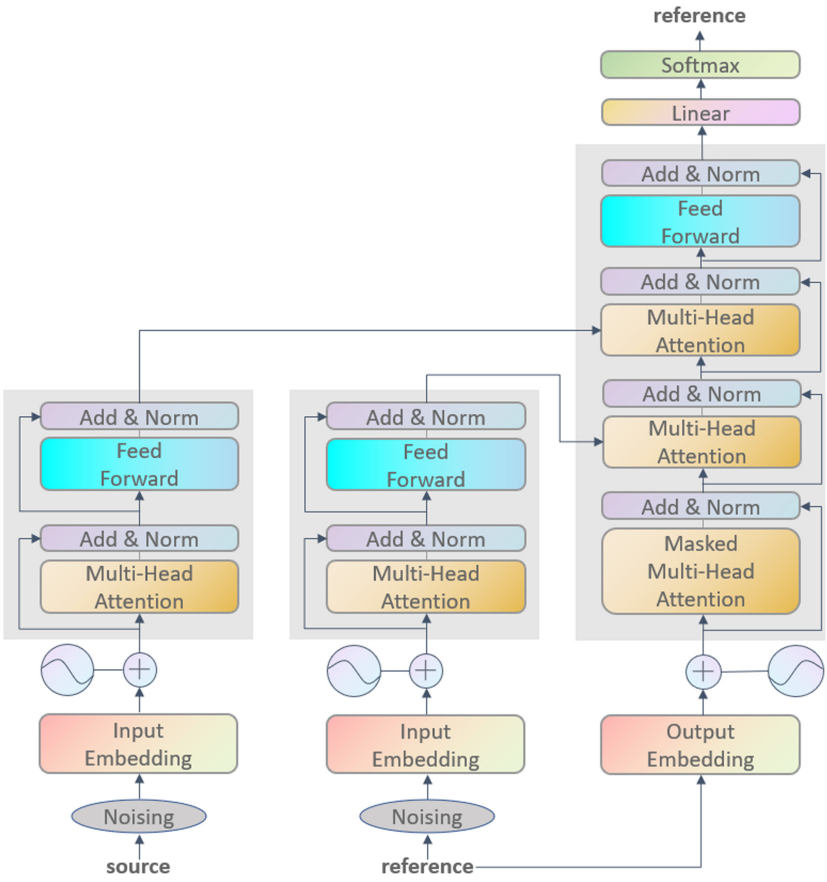

Fig. 3. Multi-source denoising autoencoder for generating synthetic triplets.

After that, the MDA can be used to generate synthetic triplets following the same formula. To be concrete, given parallel *src*-*ref* pairs, we would corrupt the *ref* by the same noising function, which is combined with *src* to generate

reconstruction via MDA. Then the original and reconstructed *ref*s are deemed as *pe* and *mt*, respectively. The generated *mt* would inevitably differ *pe* (due to the corruption-reconstruction procedure), but their connection would be close since *mt* is inferred directly from *pe*. An also because the existence of source text, the restored *mt* would be not semantically far from the *src*. This is a better simulation of the MT error distribution.

Specifically, we try the combination of three noising transformations, i.e. word omission, word replacement and word permutation. Word omission randomly omits words in a sequence, and word replacement randomly replaces words, and word permutation randomly permutes words with a maximum distance. We use the 23 million CCMT 2021 Chinese-English data, and adopt two-fold jackknifing, namely split the data into two folds, one for training and another for decoding.

However, during the experiment, we find that if the corruption on the target side is too heavy, then the model would ignore the corrupted *pe* and only attend to the *src*. In that case, our multi-source denoising autoencoder would degrade to a normal machine translation model. Therefore, we try two strategies to force the model to attend to the corrupted target text.

[1] Corrupt the source text with similar flavor;

[2] Disturbing the embedding of the source text with Gaussian noise.

Both strategies make it difficult for the autoencoder to generate reference only relying on *src*, since the information of source side is also corrupted now. Therefore, it will try to restore the target sentence by both reorganising the corrupted *pe* and translating the disturbed *src*, leading to semantically deviated (but not unrelated), and syntactically consistent *mt*.

We also follow previous works and adopt forward translation and round-trip translation to create synthetic data. Forward translation [17] uses a forward-translation model to translate *src* to the target language as *mt*. Round-trip translation [12] uses two translation models, to translate *pe* firstly to the source language then to the target language, to generate synthetic *mt*. All the translation models are trained with the 23 million data with two-fold jackknifing.

Table 7. Results on the development set of CCMT 2021 Chinese-English APE with different augmentation methods. 200k synthetic triplets is combined with 10k real triplets oversampled 20 times.

Method	Noising		Dev set	
	Source	Reference	TER	BLEU
MDA	Gaussian	Corruption	0.4016	48.44
	Corruption	Corruption	0.4023	48.41
	None	Corruption	0.4035	48.39
Forward translation	–	–	0.4039	48.41
Round-trip translation	–	–	0.4011	48.86
Ensemble	–	–	**0.3953**	**49.20**

Although the MDA-based method does not outperform the round-trip translation based method, different methods lead to different data distributions and can provide complimentary information for each other. Therefore, we use all the models for ensemble, and achieve further improvement, as shown in Table 7.

4 Conclusion

In this paper, we described our submission in CCMT 2021 quality estimation and automatic post-editing task. For QE task, we verify that the pretrained models can be further improved on target language and target domain via prefinetuning, and incorporate powerful monolingual model to perform partial-input estimation. For APE task, we find that data-scarcity is alleviated to a large extent if use pretrained model to initialize the encoder-decoder, and propose to use multi-source denoising autoencoder to generate synthetic triplets.

Due to time limitation, we only participate in the Chinese-English direction. In the future, we will extend our system to QE and APE tasks on other languages, to verify the effectiveness of our proposed methods. Besides, we will also investigate how to combine these two inner-related tasks together to achieve further improvement.

Acknowledge. The research work descried in this paper has been supported by the National Key R&D Program of China 2020AAA0108001 and the National Nature Science Foundation of China (No. 61976015, 61976016, 61876198 and 61370130). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

1. Axelrod, A., He, X., Gao, J.: Domain adaptation via pseudo in-domain data selection. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK, pp. 355–362 (2011)
2. Blatz, J., et al.: Confidence estimation for machine translation. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, Geneva, Switzerland, 23–27 Aug 2004, pp. 315–321 (2004)
3. Chatterjee, R., Negri, M., Rubino, R., Turchi, M.: 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, Belgium, Brussels, pp. 723–738 (2018)
4. Chatterjee, R., Weller, M., Negri, M., Turchi, M.: Exploring the planet of the APEs: a comparative study of state-of-the-art methods for MT automatic post-editing. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Beijing, China, pp. 156–161 (2015)
5. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, pp. 8440–8451 (2020)

6. Conneau, A., Lample, G.: Cross-lingual language model pretraining. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32 (2019)
7. Correia, G.M., Martins, A.F.T.: 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, Florence, Italy, pp. 3050–3056 (2019)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186 (2019)
9. Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., Smith, N.A.: Don't stop pretraining: Adapt language models to domains and tasks. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, pp. 8342–8360 (2020)
10. Hu, C., et al.: The NiuTrans system for the WMT20 quality estimation shared task. In: Proceedings of the Fifth Conference on Machine Translation (2020)
11. Huang, H., Xu, J., Zhu, W., Chen, Y., Dang, R.: BJTU's submission to CCMT 2020 quality estimation task. In: Li, J., Way, A. (eds.) CCMT 2020. CCIS, vol. 1328, pp. 105–113. Springer, Singapore (2020). https://doi.org/10.1007/978-981-33-6162-1_10
12. Junczys-Dowmunt, M., Grundkiewicz, R.: 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, Berlin, Germany, pp. 751–758 (2016)
13. Junczys-Dowmunt, M., Grundkiewicz, R.: 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, Belgium, Brussels, pp. 822–826 (2018)
14. Kim, H., Lee, J.H., Na, S.H.: Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In: Proceedings of the Second Conference on Machine Translation, Copenhagen, Denmark, pp. 562–568 (2017)
15. Kim, H., Lim, J.H., Kim, H.K., Na, S.H.: QE BERT: bilingual BERT using multitask learning for neural quality estimation. In: Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2), Florence, Italy, pp. 85–89 (2019)
16. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. CoRR abs/1907.11692 (2019). http://arxiv.org/abs/1907.11692
17. Negri, M., Turchi, M., Chatterjee, R., Bertoldi, N.: Escape: a large-scale synthetic corpus for automatic post-editing. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018)
18. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, pp. 223–231. Association for Machine Translation in the Americas, Cambridge (2006)
19. Specia, L., Blain, F., Logacheva, V., F. Astudillo, R., Martins, A.F.T.: Findings of the WMT 2018 shared task on quality estimation. In: Proceedings of the Third Conference on Machine Translation: Shared Task Papers, Belgium, Brussels, pp. 689–709 (2018)

20. Sun, S., Guzmán, F., Specia, L.: Are we estimating or guesstimating translation quality? In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, pp. 6262–6267 (2020)
21. Tebbifakhr, A., Agrawal, R., Negri, M., Turchi, M.: Multi-source transformer with combined losses for automatic post editing. In: Proceedings of the Third Conference on Machine Translation: Shared Task Papers, Belgium, Brussels, pp. 846–852 (2018)
22. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 30 (2017)