NJUNLP's Submission for CCMT 2022 Quality Estimation Task

Yu Zhang, Xiang Geng, Shujian Huang(✉), and Jiajun Chen

National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing, China
{zhangy,gx}@smail.nju.edu.cn, {huangsj,chenjj}@nju.edu.cn

Abstract. Quality Estimation is a task aiming to estimate the quality of translations without relying on any references. This paper describes our submission for CCMT 2022 quality estimation sentence-level task for English-to-Chinese (EN-ZH). We follow the DirectQE framework, whose target is bridging the gap between pre-training on parallel data and fine-tuning on QE data. We further combine DirectQE with the pre-trained language model XLM-RoBERTa (XLM-R) which achieves outstanding success in many NLP tasks in order to improve performance. With the purpose of better utilizing parallel data, several types of pseudo data are employed in our method as well. In addition, we also ensemble several models to promote the final results.

Keywords: Quality estimation · Pre-trained language model · DirectQE

1 Introduction

Machine translation quality estimation (QE) is the task of providing an estimate of how good or reliable the MT is without access to reference translations [15]. QE plays an important role in many real applications of machine translation. A representative example is machine translation post-editing (PE). Although the quality of machine translation for many language pairs has improved, most of the machine translations are still far from publishable quality. Therefore, a common practice for including machine translation in the workflow is to use machine translations as raw versions to be further post-edited by human translators [9]. However, post-editing low-quality machine translations spends more effort than translating from scratch [4] when QE can further improve post-editing workflows by offering more informative labels including, potentially, not only the words that are incorrect but also the types of errors that need correction [15].

Traditional QE methods make use of some hand-craft features, which are time-consuming and expensive to get [10]. Later, researchers try to generate automatic neural features by applying neural networks [1,14]. However, there are still serious problems as to the fact that QE data is scarce which limits the improvement of QE models. The Predictor-Estimator framework proposed by

T. Xiao and J. Pino (Eds.): CCMT 2022, CCIS 1671, pp. 143–150, 2022.
https://doi.org/10.1007/978-981-19-7960-6_15

Kim et al. [7] is devoted to addressing this problem, and under this framework, bilingual knowledge can be transferred from parallel data to QE tasks. The remaining drawback is that data distribution between parallel data and QE data differs. Cui et al. [2] propose the DirectQE method in order to bridge the gaps between pre-training on parallel data and fine-tuning on QE data. Nowadays, large-scale pre-trained language models have been widely applied in QE models [8], but DirectQE is not corporated with the pre-trained model which gives us insight into combining the two well-performing models.

This paper introduces our sentence-level quality estimation submission for CCMT 2022 in detail. We submit a model combining DirectQE with the pre-trained language model XLM-R for the first time. Therefore, on the one hand, the gaps between parallel data and QE data are bridged. On the other hand, the pre-trained models are well utilized in QE models. Furthermore, we try different pseudo data strategies from several aspects, including data generation and data tokenization which help us make full use of the parallel data consequently. Eventually, basic averaging ensemble and neural ensemble are used to get a better result.

2 Methods

2.1 Existing Methods

DirectQE. The DirectQE framework mainly contains two parts, the generator which is trained on parallel data to generate pseudo QE data, and the detector which can be pre-trained and fine-tuned with the pseudo data and real QE data, respectively, with the same object.

The generator of DirectQE is trained on the masked language model conditioned on source X. During the training procedure, for each parallel pair X, Y, DirectQE randomly masks 15% tokens in Y and tries to recover them. Then DirectQE predicts these masked tokens by sampling strategies according to the generated probability in the procedure of generating pseudo data. The annotating strategy is simple, it annotates the generated token as 'BAD' if it is different from the original one and the sentence-level score is the ratio of 'BAD' tokens.

The detector jointly predicts the word-level tags and sentence-level scores. It pre-trains on the pseudo QE data first and then fine-tunes on the real QE data with the same training object.

DirectQE obtained the state-of-art results when it was published.

QE BERT. QE BERT [8] uses the pre-trained model BERT (multilingual) [3] for translation quality estimation and contains two steps which are pretraining and fine-tuning separately. QE BERT further pre-trains BERT on parallel data on only the masked language model task and uses multi-task learning. In addition, the QE method based on pre-trained cross-lingual language model XLM [12] was proposed in [6].

2.2 Proposed Methods

Our proposed method contains two stages: generator and detector. The generator can be subdivided into two types, including a Transformer-based generator and an NMT-based generator. Figure 1 and Fig. 2 show the complete procedure of our methods with the Transformer-based generator and NMT-based generator separately.

Generator. The generator is trained to generate pseudo data with the use of parallel data. DirectQE adopts Transformer [16] as a generator. It functions as a word-level rewriter and is used to produce a pseudo translation with one-to-one correspondences according to the reference. In our method, we do not only adopt Transformer but also adopt a neural machine translation (NMT) as a generator, called Transformer-based generator and NMT-based generator separately. Furthermore, we try to use a Transformer-based generator to generate pseudo data at token-level and at bpe-level to eliminate the bias coming from the aspect of word tokenization.

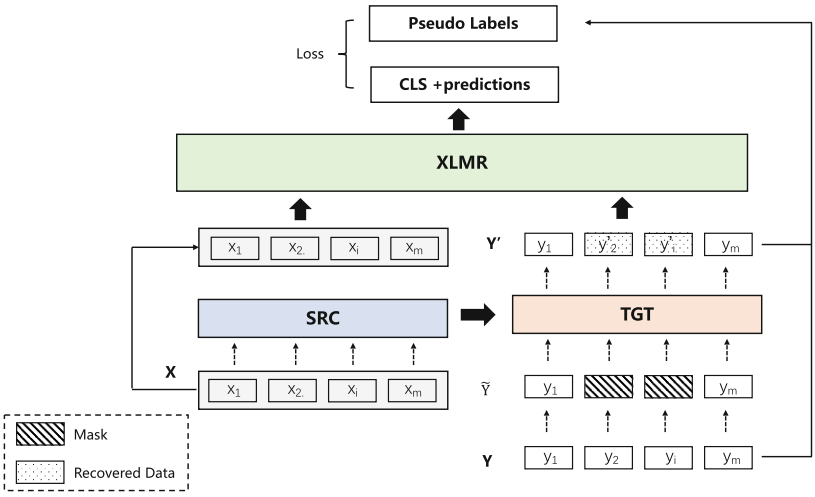

Fig. 1. Complete procedure with Transformer-based generator

Transformer-Based Generator. This part is similar to the generator in the original version of DirectQE. Given a parallel sentence pair, we randomly replace some tokens in reference with a special tag [MASK] and force the Transfomer-based generator to recover the masked tokens. Pseudo data is annotated according to the comparison between recovered tokens and their standard tokens. Different from DirectQE, we sample the tokens according to the token generation probability to better determine the location where the errors in translations most

likely occurred. Therefore, many recovered tokens may just be the same as the standard tokens causing that actual replacement ratio far below the mask ratio.

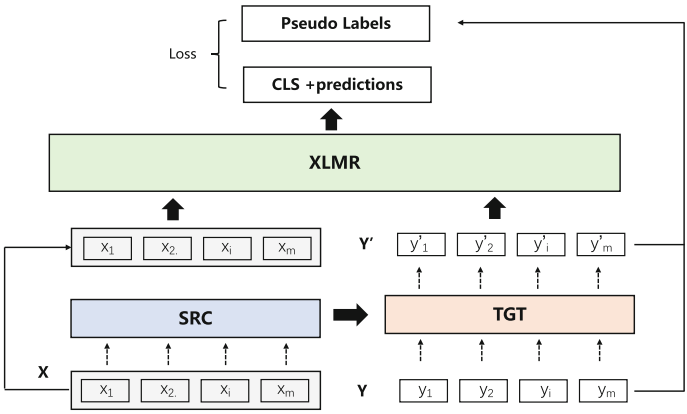

Fig. 2. Complete procedure with NMT-based generator

NMT-Based Generator. Given the parallel data, we first train a standard NMT model and use the NMT to generate target translations from the source sentences. After getting the translations, pseudo tags and scores are calculated by TERCOM [5] tool. Because the translations may be significantly different from the reference ones, the NMT-based generator is difficult to get trustworthy labels. However, this method can generate pseudo data whose distribution is consistent with real QE data that may complement the drawback of pseudo data generated from the transformer-based generator.

Detector. The detector contains two stages: pre-training and fine-tuning. Pseudo QE data generated by the generator from parallel data is used for pre-training. The pretraining task aims to jointly predict the tags O' at the word level while predicting the scores q' at the sentence level. The pretraining objectives of word-level $\mathcal{J}_w$ and sentence-level $\mathcal{J}_s$ are just the same as DirectQE:

$$\mathcal{J}_w\left(\mathbf{X}, \mathbf{Y}', o'_j\right) = \sum_{j=1}^{|\mathbf{O}'|} \log P\left(o'_j \mid \mathbf{X}, \mathbf{Y}'; \theta\right) \tag{1}$$

$$\mathcal{J}_s\left(\mathbf{X}, \mathbf{Y}', q'\right) = \log P\left(q' \mid \mathbf{X}, \mathbf{Y}'; \theta\right) \tag{2}$$

The object of the fine-tuning procedure is a little bit different from pre-training for the reason that word-level labels are not provided in real QE data. Therefore, only sentence-level scores are predicted at the stage of fine-tuning.

In DirectQE, the detector encodes the source sentence with self-attention to obtain hidden representations and predicts word-level tags from the last encoder layer at the target side as well as sentence-level scores. Different from this, the detector in our method uses the XLM-R pre-trained language model as a basic framework, while the transformer is used in DirectQE.

We concatenate the source sentence with the pseudo target sentence as a joint input. For sentence-level scores, the standard method of XLM-R uses the token corresponding to the first special token [CLS] of the last layer, and we instead combine the average representations of all the layers with the last layer representation as a mixed feature to predict the scores.

3 Experiments

In this section, we will display the details of our experiments, including the dataset, hyper-parameters, the performance of single models, and so on.

3.1 Dataset

QE Dataset. All QE triplets (SRC, MT, HTER) that we use come from the CCMT 2022 QE task, and the language direction EN-ZH that we participate in consists of 14789 training data (TRAIN) and 2826 development data (DEV).

Parallel Dataset. Parallel data is transformed into pseudo data in the form of QE triplets to pre-train the XLM-R model. We use an additional 10,000,000 out of all 20,305,269 parallel sentences from the WMT 2020 QE task and actually do not make use of parallel data provided by the CCMT QE task.

3.2 Settings

Metrics. The main metric of the quality estimation sentence-level task is Pearson's Correlation Coefficient. Mean Absolute Error (MAE) and Root Mean Squared Error(RMSE) will be considered as metrics as well.

Hyper-parameters. For the transformer-based generator, we set the mask ratio to 45% and the average HTER score of the pseudo data is approximately 16% - 18%. Except for the above, other sets are the same as the original DirectQE model. For the NMT-based generator, an inverse sqrt learning rate scheduler is used to adjust the training learning rate and set dropout to 0.3. As to the part of the detector, the XLM-R-large is used, and all the parameters are updated.

Tokenize. We first use jieba to tokenize the Chinese dataset. In the step of the generator, we use BPE [13] to tokenize both the source and target sentences, while in the step of detector SentencePiece [11] is used to tokenize the sentences for XLM-R model. The step of BPE is set to 30,000, and we use all tokens after tokenization.

3.3 Single Model Results

The results of single models are shown in Table 1. Pure-XLMR refers to the model that makes no use of both generator and parallel data but only uses real QE data. All models that, with the help of parallel data, use 3,500,000 parallel data expect that Transformed-based (10 million) uses 10 million parallel data. Similarly, all models train the model on the token level, but Transformed-based (bpe level) trains on the bpe level.

Table 1. Single model results of the CCMT 2022.

Method	Pearson	MAE	RMSE
Pure-XLMR	0.5544	0.0917	0.1367
NMT-based	0.5624	0.0901	0.1349
Transformed-based	0.5969	0.0871	0.1320
Transformed-based (10 million)	0.6138	0.0854	0.1297
Transformed-based (bpe level)	0.5847	0.0891	0.1340

It is clear that the Pure-XLMR model without parallel knowledge does not get better performance compared to other models. Meanwhile, models combining DirectQE with pre-trained language model XLM-R perform best, and with more data and with token level can get better results.

3.4 Ensemble

We try two different ensemble methods at the sentence level. The averaging ensemble is the simplest ensemble method that averages all the results from model outputs. Neural ensemble refers to that we gather all the HTERs of both training datasets and development datasets from all of the models described above. Then we train a simple neural network model that learns to use these HTER values to predict the golden HTER values.

The ensemble results are shown in Table 2, and we can see that the neural ensemble result slightly outperforms the other one at the sentence level.

Table 2. Ensemble model results of the CCMT 2022.

Ensemble method	Pearson
Averaging	0.6219
Neural result	0.6294

3.5 Analysis

In this section, we will discuss the influence of the mask ratio in the pseudo data generation procedure. We set the mask ratio to 15%, 30%, 45%, 45%*2 and 45%*3. The 45%*2 means two 45% pseudo datasets are concatenated as one pseudo data to avoid the bad effects of the high mask ratio, meanwhile, 45%*3 is similar. The corresponding average HTERs of the pseudo data are about 5%, 10%, 16%, 27%, and 36% separately. The results are shown in Fig. 3.

As we can see, the best result corresponds to 16% average HTER. Coincidentally, the average HTER of the real QE data is approximately 16% as well. The average HTERs above that get the results of a slight decrease and the average HTERs below that decline more obviously.

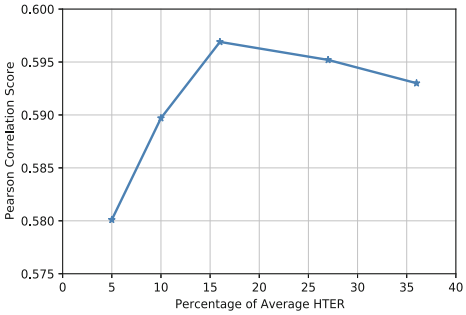

Fig. 3. QE performances according to different average HTERs

4 Conclusion

This paper describes our submissions for the CCMT 2022 Quality Estimation sentence-level task. Our systems are based on DirectQE architecture and built upon the Fairseq framework. To leverage the successful large-scale pre-trained language model, we make a combination of the high-performing DirectQE method and XLM-R pre-trained model for the first time. We also take advantage of various forms of pseudo data to better make use of parallel data for further improvements at the same time. Experiments show that the proposed method is effective. Eventually, we use base and neural ensemble methods to get our final results.

References

1. Chen, Z., et al.: Improving machine translation quality estimation with neural network features. In: Proceedings of the Second Conference on Machine Translation, pp. 551–555 (2017)

2. Cui, Q., et al.: DirectQE: direct pretraining for machine translation quality estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 12719–12727 (2021)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Escartín, C.P., Béchara, H., Orăsan, C.: Questing for quality estimation a user study. Prague Bull. Math. Linguist. **108**(1), 343–354 (2017)
5. Golden, J.P.: Terrain contour matching (TERCOM): a cruise missile guidance aid. In: Image Processing for Missile Guidance, vol. 238, pp. 10–18. SPIE (1980)
6. Kepler, F., et al.: Unbabel's participation in the WMT19 translation quality estimation shared task. arXiv preprint arXiv:1907.10352 (2019)
7. 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, pp. 562–568 (2017)
8. 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, vol. 3, pp. 85–89 (2019)
9. Koponen, M.: Is machine translation post-editing worth the effort? A survey of research into post-editing and effort. J. Spec. Transl. **25**, 131–148 (2016)
10. Kreutzer, J., Schamoni, S., Riezler, S.: Quality estimation from scratch (QUETCH): deep learning for word-level translation quality estimation. In: Proceedings of the Tenth Workshop on Statistical Machine Translation, pp. 316–322 (2015)
11. Kudo, T., Richardson, J.: SentencePiece: a simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226 (2018)
12. Lample, G., Conneau, A.: Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291 (2019)
13. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909 (2015)
14. Shah, K., Bougares, F., Barrault, L., Specia, L.: SHEF-LIUM-NN: sentence level quality estimation with neural network features. In: Proceedings of the First Conference on Machine Translation, vol. 2, Shared Task Papers, pp. 838–842 (2016)
15. Specia, L., Scarton, C., Paetzold, G.H.: Quality estimation for machine translation. Synth. Lect. Hum. Lang. Technol. **11**(1), 1–162 (2018)
16. Vaswani, A., et al.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)