Tencent Submissions for the CCMT 2020 Quality Estimation Task

Zixuan Wang, Haijiang Wu, Qingsong Ma, Xinjie Wen, Ruichen Wang, Xiaoli Wang, Yulin Zhang, Zhipeng Yao

PCG & CSIG, Tencent Inc, China
{zackiewang, harywu, qingsongma, jasonxjwen, ruichenwang, evexlwang, elwinzhang, neokevinyao}@tencent.com

Abstract. This paper presents our submissions to CCMT 2020 Quality Estimation (QE) sentence-level task for both Chinese-to-English (ZH-EN) and English-to-Chinese (EN-ZH). We propose new methods based on the predictor-estimator architecture. For the predictor, we propose XLM-predictor and Transformer-predictor. XLM-predictor novelly produces two kinds of contextual token representation, i.e., mask-XLM and non-mask-XLM. For the estimator, both RNN-estimator and Transformer-estimator are conducted and two novel strategies, i.e. top-K strategy and multi-head attention strategy, are proposed to enhance the sentence feature representation. We also propose new effective ensemble technique for sentence-level predictions.

Keywords: Quality estimation · predictor-estimator · XLM · ensemble.

1 Introduction

Machine Translation (MT) has achieved great improvement with the development of Deep Learning (DL), which requires accurate and accessible evaluation to further promote the quality of MT outputs. The widely used MT metric BLEU [7] can quickly evaluate the quality of MT outputs, on condition that the human generated reference translation is provided in advance. However, high-quality reference translations demand labor and time. Quality Estimation (QE) becomes an alternative method to evaluate the quality of MT outputs with no access to reference translations [2, 11].

Our submissions focus on the sentence-level sub-task of the CCMT 2020 QE Shared Task in both English-to-Chinese (EN-ZH) and Chinese-to-English (ZH-EN) directions. The sentence-level task aims to predict the Human-targeted Translation Edit Rate (HTER) [8] of the MT output, which reflects the minimal amount of edits that is needed to process the MT output to an acceptable level, thus denotes the overall quality of the MT output.

Sentence-level QE is commonly formulated as a regression problem. The classical baseline model QuEst++ [9] constructed rule-based features and employed machine learning algorithm to predict HTER scores. Recent neural networks applied the newly-emerged predictor-estimator architecture to QE tasks. Kim

et al. [5] proposed the predictor-estimator model first. The predictor aims to extract feature vectors by incorporating large parallel data into a bilingual RNN model, which is subsequently fed into the main bidirectional RNN model to predict QE scores. Later, Fan et al. [1] replaced the RNN-based predictor with a bidirectional Transformer and added 4-dimensional mis-matching features. NiuTrans[10] used Transformer-DLCL based predictor, whereas Unbabel [12] introduced BERT and XLM pretrained predictor models. Besides, ensemble technique emerges as a new trend that can further improve the QE performance. The ensemble approach achieved the best results in the sentence-level QE sub-task of both CCMT 2019 [11] and WMT 2019 [2].

We submit a predictor-estimator based QE system, which extends the open-source OpenKiwi framework[1] [4] to take advantage of recently proposed pretrained models by transferring learning techniques. Our contributions are as follows:

- We implement two predictors as feature extractors, the Transformer-based predictor (Transformer-predictor) [1], and the XLM-based predictor (XLM-predictor) [6] via the transfer learning technique. For XLM-based predictor, it produces two kinds of contextual token representation in a novel fashion, i.e., masked representations and non-masked representations.
- In addition to the LSTM-based estimator (LSTM-estimator), we use transformer neural networks to build a Transformer-based estimator (Transformer-estimator). We propose novel strategies to optimize the sentence features, i.e., top-K strategy and multi-head attention strategy.
- We ensemble several single-models by regression algorithms to produce a single sentence-level prediction, which outperforms the commonly-used arithmetic average.

2 Architecture

We employ the predictor-estimator architecture built upon the OpenKiwi framework. We adopt XLM-predictor and Transformer-predictor respectively to extract contextual feature vector of the MT output, which could reflect semantic information between the source and the MT output. We innovatively propose mask-XLM and non-mask-XLM, which will be demonstrated in detail below. For the estimator, similarly, different models are used. We adopt LSTM-estimator and Transformer-estimator. Two effective sentence representation strategies for LSTM-estimator are proposed.

2.1 Predictors

2.1.1 XLM-Predictor

[1] https://github.com/Unbabel/OpenKiwi

The Cross-lingual Language Model (XLM) achieved state-of-the-art performances on several Natural Language Preprocessing (NLP) tasks [6]. We extend XLM to QE task and propose novel XLM-predictor.

First, we fine-tune XLM with both Masked Language Modeling (MLM) and Translation Language Modeling (TLM) using large-scale parallel data following the XLM instructions [2].

Instead of using target word representations produced by the fine-tuned XLM as the predictor output as in Kepler et al. [12], we propose non-mask-XLM representation and mask-XLM representation, and adopt further computation to enhance the feature ability. For non-mask-XLM, all words are fed into the XLM to predict each word's representation, enabling the word itself to help predict its representation. For mask-XLM, one target word is masked one time so that the prediction of the masked target word leverages only the surrounding target words and the source context, without any prior information from itself. Let the length of the target sentence be N, the mask-XLM process is repeated N times and thus all target word representations are generated. We further consider two aspects that influence the word representation. One is the weight of each dimension in the word representation. We continue to use the weight during the fully connected layers in XLM. The other is the language embedding, considering that the word representation is closely related to the corresponding language. Formula 1 describes the final word representation produced by XLM-predictor, which is then fed into the estimator as input features to predict HTER scores.

$$Rep_i = R_i \cdot (W_i + Emb_{lang}) \tag{1}$$

where i refers to the i-th word in the target sentence, R_i refers to the original representation of the i-th word, W_i and Emb_{lang} denote the weight of the i-th word and the language embedding of the target sentence respectively. Rep_i is the final representation of the i-th word.

2.1.2 Transformer-Predictor

Transformer-predictor has been proved effective by Fan et al. [1]. Our predictor follows their bidirectional transformer, which contains three modules: self-attention for the source sentence, forward and backward self-attention encoders for the target sentence, and the re-constructor for the target sentence. The semantic features are extracted by bidirectional transformer and human-crafted mismatching features. Our predictor has made one modification: multi-decoding is used during the machine translation module.

To improve the performance, we integrate a XLM-based model, which simply replace the predictor part by XLM. We take the weighted average the two models as the final sentence-level prediction as shown in formula 2. We set α as 0.8 since we emphasize the transformer-based predictor's contribution and incorporate

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

XLM-based predictor only to further enhance the overall performance.

$$Score = \alpha * Score_{Transformer} + (1 - \alpha) * Score_{XLM} \tag{2}$$

2.2 Estimators

Estimator takes features produced by predictor as the input to predict sentence-level scores of the MT output. We implement a multi-layer LSTM-estimator and a Transformer-estimator, both of which adopt novel strategies to optimize the sentence features.

The last state or the the mean pooling of hidden states are usually taken as the sentence representation. However, they both have weaknesses: the last state losses certain information of the whole sentence due to the information decay problem, while the mean pooling distributes the same weights to all hidden states. Actually, the contribution of each word to the sentence features varies, which inspires us to take the concept of weight into consideration. We propose two strategies, top-K strategy and multi-head attention strategy, which computing weights from two different perspectives. The two strategies are illustrated in Figure 1.

2.2.1 Top-K Strategy

Each hidden state is a word representation vector, and each element of the vector represents one feature dimension. From feature dimension perspective, Top-K strategy forms the sentence features by concatenating top-K elements of each feature dimension. The top-K elements refer to the top-K values among all words of the current focus feature dimension. In a result, the sentence feature is a vector with size K * number of feature dimensions.

2.2.2 Multi-head Attention Strategy

Different from top-K strategy, multi-head attention strategy considers the impact of each word on the sentence features via attention mechanism. For each head, we obtain a vector which is a weighted sum of all the word features. By repeating K times, the final sentence feature is a vector with size K * number of feature dimensions. We demonstrate the computation process as formula 3 and 4,

$$a_{k_i} = softmax(h_i * W_k), \tag{3}$$

$$f_{sent} = [\sum_i \alpha_{1_i} * h_i, \ldots, \sum_i \alpha_{k_i} * h_i] \tag{4}$$

where a_{k_i} is attention results of each word (h_i), and f_{sent} is the final sentence feature representation.

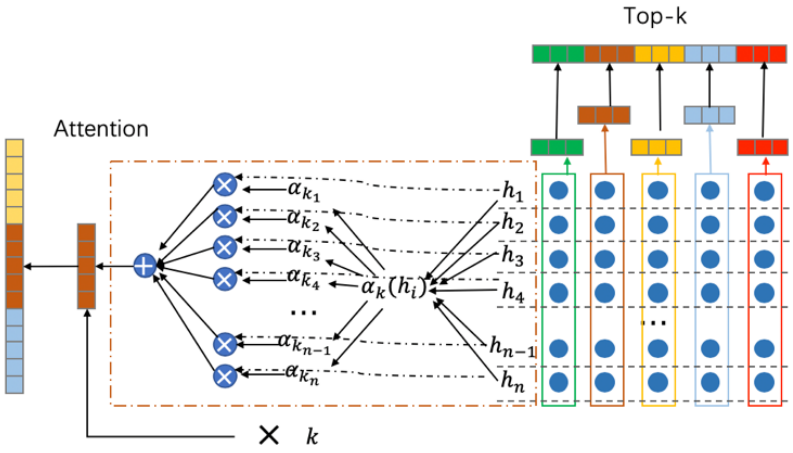

Fig. 1. Sentence Representation Strategies.

2.3 Ensemble

To boost performance, we ensemble several systems to produce a single sentence score prediction. Model stacking [13, 14] is an efficient ensemble method in which the predictions, generated by using various single systems, are used as inputs of regression algorithm implemented within a two-layer model. To avoid overfitting, we use k-fold cross validation and set k = 5, as described in Martin et al. [15].

We implement and compare several regression algothrims, i.e. Powell's method [16], Quantile Regression, Support Vector Regression (SVR) and Logistic Regression (LR) to optimize for the task metric - Pearson correlation.

3 Experiments and Results

The experiment details below refer to the CCMT 2020 sentence-level QE task only.

3.1 Dataset

The dataset consists of parallel data and QE triplets. Parallel data is used to train the predictor to produce contextual features, which is provided by the CCMT QE task with 8,023,011 EN-ZH parallel sentences (Repeated parallel sentences are filtered). Besides, we use additional 37,128,402 parallel sentences from WMT 2020 task. QE triplets (src, mt, hter) are provided by CCMT QE

task, consisting of 10,070 training data (TRAIN) and 1,143 development data (DEV) for ZH-EN, and 14,789 training data and 1,381 DEV for EN-ZH.

We correct one abnormal detail in both the QE TRAIN and DEV triplets for ZH-EN. Take the following sentence as an example: *"Our position is to be courageous, step to be stable.We should not only explore boldly, but also be reliable and prudent, thinking twice before act."*

Two English words are connected by a full stop punctuation without any white-space in the machine translation (MT) file and the post-edited (PE) file. This phenomenon hardly appears in the machine translation and will lead to two possible problems. One is the correctness of HTER scores, which are the gold scores for the training process of QE systems. On the other hand, it will increase Unknown words (UNK), which may exert negative effects on the performance of QE systems. We therefore add white-space between two connected words and re-compute HTER scores according to the official scripts.

3.2 Experiments

3.2.1 Experiments with the XLM-Predictor

For the XLM-predictor, we experiment non-mask-XLM predictor ($non-mask$) and mask-XLM ($mask$) predictor respectively. We also try to concatenate feature vectors produced from the two predictors ($Both$) as the input for the next estimator procedure. Fixing the XLM-predictor, we conduct experiments with LSTM-estimator ($LSTM$) and Transformer-estimator (TF), each of which adopts multi-head attention strategy ($attn$) or top-K strategies ($topK$) to improve the sentence representation.

The results in Table 1 show that our QE systems with XLM predictor achieve moderate correlation with HTER scores in general. On ZH-EN, mask_LSTM_topK ranks top with a Pearson score of .5690, whereas the non-mask_LSTM_attn ranks top with .5329 on EN-ZH. The language features could be an explanation why non-mask-XLM performs better than mask-XLM for Chinese: The Chinese word meaning usually different from that of the consisting characters, because Chinese word meaning is not the simple addition of the consisting characters.

3.2.2 Experiments with the Transformer-Predictor

We implement a Transformer-based predictor-estimator following Fan et al. [1]. Transformer-predictor has one modification, i.e. the use of multi-decoding during machine translation. To further improve the overall performance, XLM-based predictor is incorporated but with a smaller weight compared to transformer-based predictor as describe in Section 2.1.2.

Experiments with the Transformer-Predictor are shown in Table 2, which presents both key configurations and results.

Table 1. Pearson correlations of single QE systems with XLM-Predictor on CCMT 2020 QE EN-ZH and ZH-EN development set for sentence-level task.

Model	**ZH-EN**	**EN-ZH**
Both_LSTM_attn	.5468	.5244
Both_LSTM_topK	.5620	.5205
Both_TF_attn	.5364	.4865
Both_TF_topK	.5350	.5056
mask_LSTM_attn	.5542	.4982
mask_LSTM_topK	**.5690**	.4956
mask_TF_attn	.5540	.4951
mask_TF_topK	.5603	.4978
non-mask_LSTM_attn	.5365	**.5329**
non-mask_LSTM_topK	.5507	.5277
non-mask_TF_attn	.5345	.5179
non-mask_TF_topK	.5382	.5208

Table 2. Pearson correlations of single QE systems with Transformer-Predictor on CCMT 2020 QE EN-ZH and ZH-EN development set for sentence-level task.

	Model	**Model2**	**Model3**	**Model4**	**Model5**
XLM-EST-dim	5140	5140	5140	0	0
Trans-EST-dim	5140	5140	5140	5140	5140
XLM_finetune	1	1	0	1	1
XLM-tgt-only	0	1	1	1	1
EST-hidden-dim	512	256	256	256	512
Pearson-ZH-EN	**.549**	.547	**.549**	.512	.51
Pearson-EN-ZH	.491	**.495**	.491	.456	.453

In table 2, XLM-EST-dim means the dimension in fully connected layer of estimator in XLM-based predictor, while Trans-EST-dim means that in transformer-based predictor. XLM_finetune denotes whether XLM is fine-tuned and XLM-tgt-only means only target information is used in XLM. EST-hidden-dim is the hidden dimension in estimator.

3.2.3 Experiments with ensemble methods

We conduct multiple single QE systems through different model architectures or the same architecture with different parameters, and integrate the predictions via stacking ensemble with 4 regressors respectively.

We select 24 systems based on XLM-predictor and 5 based on Transformer-predictor, then filter single systems with a Pearson score less than 0.5 during ensembling, which leads to 13 systems for EN-ZH, 12 systems for ZH-EN on DEV and 11 system for ZH-EN on PSEU_DEV respectively. 4 regressors refer to Powell's, Quantile Regression, SVR and LR.

Results on DEV with filtered systems are shown in Table 3 prove the effectiveness of ensemble, compared with results shown in Table 1 and Table 2. From Table 3, we also conclude that regression algorithms outperform the simple averaging of single system predictions ("Average" in Table 3).

Table 3. Pearson correlations of ensemble QE systems on CCMT 2020 QE EN-ZH and ZH-EN development set for sentence-level task.

Ensemble methods	**ZH-EN**	**EN-ZH**
Average	.5648	.5408
Powell's	.5839	**.5592**
Quantile Regression	**.5848**	.5530
SVR	.5643	.5449
LR	.5843	.5588

4 Conclusion

We describe our submissions to CCMT 2020 QE sentence-level task. Our systems are based on predictor-estimator architecture and built upon OpenKiwi framework. We implement two predictors, Transformer-predictor and XLM-predictor. XLM-predictor novelly produces two kinds of contextual token representation, i.e., masked representations and non-masked representations. Both RNN-estimator and Transformer-estimator are conducted to predict the MT output scores by using the features produced from predictor. Two novel strategies, i.e. top-K strategy and multi-head attention strategy, are proposed to enhance the sentence feature representation. Stacking ensemble is also proved to be more effective than simple averaging integration.

References

1. Fan, K., Wang, J., Li, B., Zhou, F., Chen, B., Si, L.: "Bilingual Expert" Can Find Translation Errors. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6367-6374 (2019)
2. Fonseca, E., Yankovskaya, L., Martins, A.F., Fishel, M., Federmann, C.: Findings of the WMT 2019 Shared Tasks on Quality Estimation. In: Proceedings of the Fourth Conference on Machine Translation, vol. 3, pp. 1-10. ACL, Florence (2019)
3. Kepler, F., Trénous, J., Treviso, M., Vera, M., Góis, A., Farajian, M.A., Lopes, A.V., Martins, A.F.: Unbabel's Participation in the WMT19 Translation Quality Estimation Shared Task. In: Proceedings of the Fourth Conference on Machine Translation, pp. 78-84. ACL, Florence (2019)
4. Kepler, F., Trénous, J., Treviso, M., Vera, M., Martins, A.F.: OpenKiwi: An Open Source Framework for Quality Estimation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 117-122. ACL, Florence (2019)
5. Kim, H., Jung, H.Y., Kwon, H., Lee, J.H., Na, S.H.: Predictor-Estimator: Neural Quality Estimation based on Target Word Prediction for Machine Translation. In: ACM Transactions on Asian and Low-Resource Language Information Processing, 17(1), pp. 1-22 (2017)
6. Lample, G., Conneau, A.: Cross-lingual Language Model Pretraining. In: Advances in Neural Information Processing Systems 32, pp. 7059–7069. NeurIPS, Vancouver (2019)
7. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311-318. ACL, Philadelphia (2002)
8. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A Study of Translation Edit Rate with Targeted Human Annotation. In: Proceedings of Association for Machine Translation in the Americas, pp. 223-231. AMTA, Cambridge (2006)
9. Specia, L., Paetzold, G., Scarton, C.: Multi-level Translation Quality Prediction with QuEst++. In: Proceedings of ACL-IJCNLP 2015 System Demonstrations, pp. 115-120. ACL-IJCNLP, Beijing (2015)
10. Wang, Z., Liu, H., Chen, H., Feng, K., Wang, Z., Li, B., Xu, C., Xiao, T., Zhu, J.: NiuTrans Submission for CCMT19 Quality Estimation Task. In: China Conference on Machine Translation, pp. 82-92. Springer, Singapore (2019)
11. Yang, M., Hu, X., Xiong, H., Wang, J., Jiaermuhamaiti, Y., He, Z., Luo, W., Huang, S.: CCMT 2019 Machine Translation Evaluation Report. In: China Conference on Machine Translation, pp. 105-128. Springer, Singapore (2019)
12. Kepler F, Trénous J, Treviso M, et al. Unbabel's Participation in the WMT19 Translation Quality Estimation Shared Task[J]. arXiv preprint arXiv:1907.10352, 2019.
13. Wolpert D H. Stacked generalization[J]. Neural networks, 1992, 5(2): 241-259.
14. Breiman L. Stacked regressions[J]. Machine learning, 1996, 24(1): 49-64.
15. Martins A F T, Junczys-Dowmunt M, Kepler F N, et al. Pushing the limits of translation quality estimation[J]. Transactions of the Association for Computational Linguistics, 2017, 5: 205-218.
16. Powell M J D. An efficient method for finding the minimum of a function of several variables without calculating derivatives[J]. The computer journal, 1964, 7(2): 155-162.