Tencent submission for WMT20 Quality Estimation Shared Task

Haijiang Wu Zixuan Wang Qingsong Ma Xinjie Wen
Ruichen Wang Xiaoli Wang Yulin Zhang Zhipeng Yao Siyao Peng
PCG & CSIG, Tencent Inc, China
{harywu, zackiewang, qingsongma, jasonxjwen, ruichenwang,
evexlwang, elwinzhang, neokevinyao, logansypeng}@tencent.com

Abstract

This paper presents Tencent's submission to the WMT20 Quality Estimation (QE) Shared Task: Sentence-Level Post-editing Effort for English-Chinese in Task 2. Our system ensembles two architectures, XLM-based and Transformer-based Predictor-Estimator models. For the XLM-based Predictor-Estimator architecture, the predictor produces two types of contextualized token representations, i.e., masked XLM and non-masked XLM; the LSTM-estimator and Transformer-estimator employ two effective strategies, top-K and multi-head attention, to enhance the sentence feature representation. For Transformer-based Predictor-Estimator architecture, we improve a top-performing model by conducting three modifications: using multi-decoding in machine translation module, creating a new model by replacing the transformer-based predictor with XLM-based predictor, and finally integrating two models by a weighted average. Our submission achieves a Pearson correlation of 0.664, ranking first (tied) on English-Chinese (Specia et al., 2020).

1 Introduction

The development of Machine Translation (MT) requires efficient quality evaluation of the outputs. The widely used MT metric BLEU (Papineni et al., 2002) satisfies this demand. However, BLEU requires human reference translations which costs labor and time to generate. Quality Estimation (QE) is an alternative to evaluate the quality of MT outputs with no access to reference translations (Fonseca et al., 2019; Yang et al., 2019).

We participate in the sentence-level task in Task 2 of the WMT20 QE Shared Task for English–Chinese (Specia et al., 2020). The sentence-level task aims to predict the Human-targeted Translation Edit Rate (HTER) (Snover et al., 2006) of the MT output, which reflects the minimal amount of edits that is needed to post-edit the MT output to an acceptable one, thus denotes the overall quality of the MT output.

The classical baseline model QuEst++ (Specia et al., 2015) 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. (2017) proposed the first predictor-estimator model to extract feature vectors by incorporating large parallel data into a bilingual RNN model, which is subsequently fed into another bidirectional RNN model to predict QE scores. Later on, Fan et al. (2019) replaced the RNN-based predictor by a bidirectional Transformer and added 4-dimensional mis-matching features; Wang et al. (2019) used a Transformer-DLCL based predictor; and Kepler et al. (2019a) introduced BERT and XLM pretrained predictors into their system. Besides the improvements on model architectures, choosing the top-performing models using ensemble techniques further improves the QE performance. For example, the submission using ensemble techniques achieved the best result in the sentence-level QE sub-task in both WMT19 (Fonseca et al., 2019) and CCMT19 (Yang et al., 2019).

We submit a predictor-estimator based QE system, which extends the open-source OpenKiwi framework[1] (Kepler et al., 2019b) to take advantage of recently proposed pre-trained models by transfer learning technique. Our contributions are as follow:

- We propose XLM-based Predictor-Estimator architecture, which introduces the cross-lingual language model (XLM) (Lample and Conneau, 2019) to QE task via transfer learning technique. Instead of directly using target

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

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1062–1067
Online, November 19–20, 2020.

word representations produced by XLM as the predictor output, we propose non-masked XLM representation and masked XLM representation, and adopt further computation to enhance the feature extraction ability.

- We implement LSTM-based estimator and Transformer-based estimator, with two novel strategies to enhance the sentence feature representation, i.e. top-K strategy and multi-head attention strategy.

- We reform Transformer-based Predictor-Estimator (Fan et al., 2019) by using multi-decoding during the machine translation module. Besides, we create a new model by replacing the transformer-based predictor with XLM-based predictor, and then integrate the two models by weighted average.

- We ensemble several single-models by regression algorithms to produce a single sentence-level prediction, which outperforms the commonly-used arithmetic average.

We next describe the models, experiments and results in detail.

2 Models

Our models employ predictor-estimator architecture and OpenKiwi framework. Overall, we implement two predictor-estimator architectures, namely XLM-based Predictor-Estimator and Transformer-based Predictor-Estimator, and ensemble multiple systems to boost performance.

2.1 XLM-based Predictor-Estimator

XLM achieved state-of-the-art performances on several NLP tasks (Lample and Conneau, 2019). We extend XLM by transferring the language model to QE task and proposing a novel XLM-based Predictor-Estimator model.

2.1.1 Predictor

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

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

XLM representations Instead of using target word representation produced by the fine-tuned XLM as the predictor output as in Kepler et al. (2019a), we propose non-masked XLM representation and masked XLM representation, and adopt further computation to enhance the feature extraction ability. For non-masked XLM, all words are fed into the XLM to predict each word's representation, letting the word itself help to predict its representation. For masked XLM, one target word is masked one time so that the prediction of the masked word leverages information only from the surrounding words and the source context, without any prior information from itself.

Let the length of the target sentence be N, the masking process is repeated N times and then all target word representations are generated. We consider two aspects that influence word representation: the weight of each dimension in the word representation and the language embeddings. Formula (1) describes the final word representation, which is then fed into the estimator as input features to predict HTER scores.

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

In formula (1), i refers to the i-th word in the target sentence and R_i refers to the original representation of the i-th word. W_i denotes the weights of each dimension for the i-th word and Emb_{lang} denotes the language embedding of the target sentence. Rep_i is the final representation of the i-th word.

2.1.2 Estimator

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

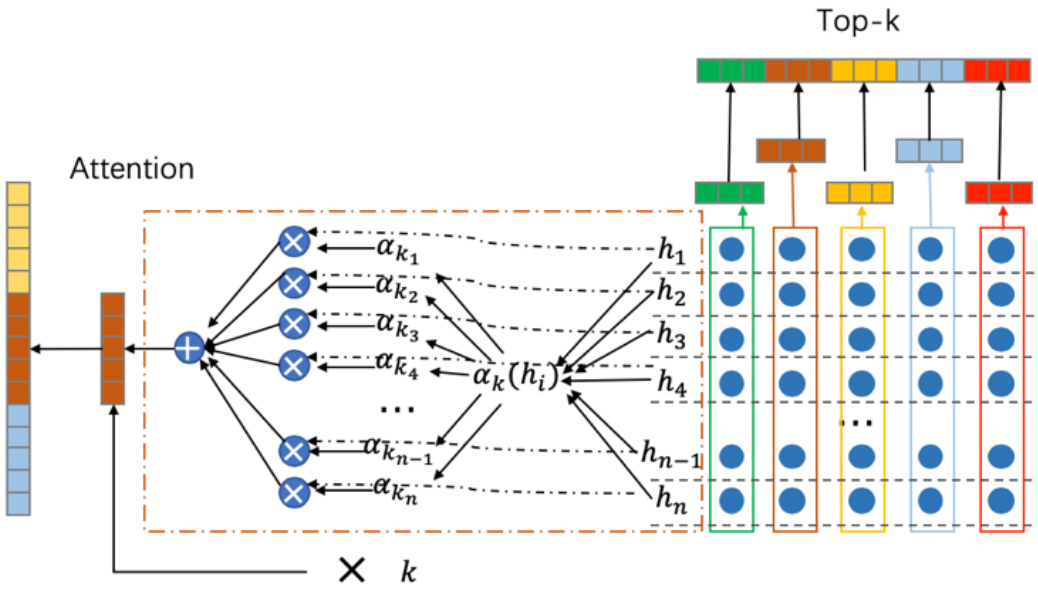

Figure 1: Top-K strategy and Multi-head attention strategy illustration.

two different perspectives, as shown in Figure 1.

Top-K Strategy The hidden state of each word is a vector, and each element of the vector represents one feature dimension. The top-K strategy forms sentence features by concatenating top K elements of each of N feature dimension, creating a vector of size K * N.

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 * N. We demonstrate the computation process as in Formula (2) and (3),

$$a_{k_i} = softmax(h_i * W_k), \tag{2}$$

$$f_{sent} = [\sum_i a_{1_i} * h_i, \ldots, \sum_i a_{k_i} * h_i] \tag{3}$$

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

2.2 Transformer-based Predictor-Estimator

Transformer-based Predictor-Estimator architecture has been proved effective by Fan et al. (2019). 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. We include semantic features extracted by bidirectional transformer and human-crafted mismatching features in the model. Our Transformer-based model has three main improvements:

- For transformer-based predictor, we use multi-decoding during the machine translation module.
- We create a XLM-based predictor, which simply replace the predictor part by XLM.
- We take the weighted average of the two models as the final sentence-level prediction as shown in Formula (4). We set α to be 0.8 since the transformer-based predictor contributes more than the XLM-based predictor.

$$Score = \alpha * Score_{Transformer} + (1 - \alpha) * Score_{XLM} \tag{4}$$

2.3 Ensemble

To boost performance, we ensemble several systems to produce a single sentence score prediction. Model stacking (Wolpert, 1992; Breiman, 1996) is an efficient ensemble method in which the predictions, generated by using various single systems, are used as inputs in a second-layer regression algorithm. To avoid over-fitting, we use k-fold cross validation with k = 5 (Martins et al., 2017).

We implement and compare several regression algothrims, i.e., Powell's method (Powell, 1964), Quantile Regression, Support Vector Regression (SVR), and Logistic Regression (LR) to optimize the task on Pearson correlation.

3 Experiments and Results

We conducted three sets of experiments on the WMT20 QE English-Chinese Sentence-level Task in Task 2.

3.1 Dataset

The dataset consists of parallel data between English and Chinese, as well as QE triplets with source texts, target translations and HTER scores. The parallel data is used to train the predictor to produce contextualized features. Specifically, we sampled 45M English-Chinese parallel sentences to train the XLM-based Predictor. For Transformer-based Predictor, we combined the subset of 8.9M parallel sentences in CCMT20 with a set of 15K pseudo data constructed by augmenting the number of entities within the sentences.

3.2 Experiments

3.2.1 Experiments with XLM-based Predictor-Estimator

We experiment with non-masked ($non\text{-}masked$) and masked XLM ($masked$) predictors. We also try to concatenate feature vectors produced by two predictors ($Both$) as the input of the next estimator procedure. For every 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 strong correlation with HTER scores in general. The model with both predictors, LSTM-estimator and multi-head attention strategy (*Both_LSTM_attn*) ranks top with a Pearson score of .6348.

Model	Pearson
Both_LSTM_attn	**.6348**
Both_LSTM_topK	.6244
Both_TF_attn	.6218
Both_TF_topK	.6276
masked_LSTM_attn	.6232
masked_LSTM_topK	6156
masked_TF_attn	.6143
masked_TF_topK	.6260
non-masked_LSTM_attn	.6142
non-masked_LSTM_topK	.6216
non-masked_TF_attn	.6234
non-masked_TF_topK	.6268

Table 1: Pearson correlations of single QE systems with XLM-based Predictor-Estimator on WMT20 English-Chinese development set for sentence-level task.

3.2.2 Experiments with Transformer-based Predictor-Estimator

We extend Transformer-based predictor-estimator (Fan et al., 2019) with the following modifications: we use multi-decoding during Transformer-based predictor, replace Transformer-based predictor with XLM-based predictor to form a new model, and then integrate the two models into one by weighted average with more weights on the Transformer-based predictor.

Table 2 presents the key configurations and results in Transformer-based experiments. Among the four models, Models 1–3 integrate XLM-based estimators into the architecture and achieve the highest Pearson correlations of .646–.647. These integrated models vary in two configurations: whether or not the XLM-estimator has been fine-tuned and whether or not to include source information. We conclude that XLM-based model helps improve Transformer-based Predictor-Estimator performance.

3.2.3 Experiments with ensemble methods

We conduct multiple single QE systems through different model architectures or the same architecture with different parameters. Specifically, we include predictions from 24 XLM-based and 5 Transformer-based Predictor-Estimator systems, and stack them using 4 regressors: Powell's, Quan-

	Transformer	XLM Estimator			Pearson
		Included?	Finetuning?	Input	
Model 1	✔	✔	✔	source & target	**.646**
Model 2	✔	✔	✔	target only	**.647**
Model 3	✔	✔	✘	target only	**.647**
Model 4	✔	✘	N/A	N/A	.633

Table 2: Pearson correlations of single QE systems with Transformer-based Predictor-Estimator on WMT20 English-Chinese development set for sentence-level task.

tile Regression, SVR and LR.

Results in Table 3 prove the effectiveness of ensemble techniques, when compared to results shown in Tables 1 and 2. We also conclude that regression algorithms outperform simple averaging ("Average" in Table 3), and among them, Logistic Regression achieves the best Pearson score of .6785.

Ensemble methods	**Pearson**
Average	.6521
Powell's	.6515
Quantile Regression	.6699
Support Vector Regression	.6735
Logistic Regression	**.6785**

Table 3: Pearson correlations of ensemble QE systems on WMT20 English-Chinese development set for sentence-level task.

4 Conclusion

We describe Tencent's submission to the WMT20 Quality Estimation sentence-level task in task 2. Our systems are based on predictor-estimator architecture and built upon OpenKiwi framework. We implement two predictor-estimator architectures, XLM-based Predictor-Estimator and Transformer-based Predictor-Estimator. For XLM-based Predictor-Estimator, we produces two kinds of contextual token representation, masked and non-masked representations. Both LSTM-estimator and Transformer-estimator are conducted to predict the MT output scores by using the features produced from the predictors. Top-K strategy and multi-head attention strategy are employed to enhance the sentence feature representation. For Transformer-based Predictor-Estimator, we integrate one model based on XLM-based predictor to enhance the overall performance. Stacking ensemble is also proved to be more effective than simple averaging integration.

References

Leo Breiman. 1996. Stacked regressions. *Machine learning*, 24(1):49–64.

Kai Fan, Jiayi Wang, Bo Li, Fengming Zhou, Boxing Chen, and Luo Si. 2019. "bilingual expert" can find translation errors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6367–6374.

Erick Fonseca, Lisa Yankovskaya, André FT Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the wmt 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10.

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M Amin Farajian, António V Lopes, and André FT Martins. 2019a. Unbabel's participation in the wmt19 translation quality estimation shared task. *arXiv preprint arXiv:1907.10352*.

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André FT Martins. 2019b. Openkiwi: An open source framework for quality estimation. *arXiv preprint arXiv:1902.08646*.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(1):1–22.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

André FT Martins, Marcin Junczys-Dowmunt, Fabio N Kepler, Ramón Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics*, 5:205–218.

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

40th annual meeting of the Association for Computational Linguistics, pages 311–318.

Michael JD Powell. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120.

Ziyang Wang, Hui Liu, Hexuan Chen, Kai Feng, Zeyang Wang, Bei Li, Chen Xu, Tong Xiao, and Jingbo Zhu. 2019. Niutrans submission for ccmt19 quality estimation task. In *China Conference on Machine Translation*, pages 82–92. Springer.

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.

Muyun Yang, Xixin Hu, Hao Xiong, Jiayi Wang, Yiliyaer Jiaermuhamaiti, Zhongjun He, Weihua Luo, and Shujian Huang. 2019. Ccmt 2019 machine translation evaluation report. In *China Conference on Machine Translation*, pages 105–128. Springer.