CrossQE: HW-TSC 2022 Submission for the Quality Estimation Shared Task

Shimin Tao\textsuperscript{1*}, Chang Su\textsuperscript{1}, Miaomiao Ma\textsuperscript{1}, Hao Yang\textsuperscript{1}, Min Zhang\textsuperscript{1}, Xiang Geng\textsuperscript{2}, Shujian Huang\textsuperscript{2}, Jiaxin Guo\textsuperscript{1}, Minghan Wang\textsuperscript{1}, Yinglu Li\textsuperscript{1}
\textsuperscript{1}Huawei Translation Services Center, China
\textsuperscript{2}Nanjing University, China
\{taoshimin, suchang8, mamiaomiao, yanghao30, zhangmin186, guojiaxin1, wangminghan, liyinglu\}@huawei.com
gx@smail.nju.edu.cn, huangsj@nju.edu.cn

Abstract

Quality estimation (QE) is a crucial method to investigate automatic methods for estimating the quality of machine translation results without reference translations. This paper presents Huawei Translation Services Center’s (HW-TSC’s) work called CrossQE in WMT 2022 QE shared tasks 1 and 2, namely sentence- and word-level quality prediction and explainable QE. CrossQE employs the framework of predictor-estimator for task 1, concretely with a pre-trained cross-lingual XLM-RoBERTa large as predictor and task-specific classifier or regressor as estimator. An extensive set of experimental results show that after adding bottleneck adapter layer, mean teacher loss, masked language modeling task loss and MC dropout methods in CrossQE, the performance has improved to a certain extent. For task 2, CrossQE calculated the cosine similarity between each word feature in the target and each word feature in the source by task 1 sentence-level QE system’s predictor, and used the inverse value of maximum similarity between each word in the target and the source as the word translation error risk value. Moreover, CrossQE has outstanding performance on QE test sets of WMT 2022.

1 Introduction

Quality estimation (QE) is the task of evaluating a translation system’s quality without access to reference translations (Specia et al., 2018). In WMT 2022 QE shared task \textsuperscript{1}, there are three tasks — Quality Prediction, Explainable QE and Critical Error Detection. Each task involves several language pairs. Our team — Huawei Translation Services Center (HW-TSC) — participated in quality prediction and explainable QE tasks over all language pairs.

This paper describes the HW-TSC’s systems called CrossQE submitted for these tasks. Some key steps are summarized as follow:

- We used pre-trained Cross-lingual XLM-Roberta large (Lample and Conneau, 2019; Conneau et al., 2019) as predictor instead of RNN-based model in the two-stage Predictor-Estimator architecture (Kim et al., 2017). The task-specific classifier or regressor is used as quality estimator, and multitasks are trained at the same time.

- The cross-lingual XLM-RoBERTa large model is pre-trained on large-scale parallel corpora where source and target tokens are concatenated by MLM task. Shuffling those tokens and predicting those tokens’ index by the pre-trained model as an additional pre-training task can improve the QE model’s effect.

- We build on the COMET architecture \textsuperscript{2} by exploring adapter layers (Houlsby et al., 2019) for quality estimation to eliminate the overfitting problem instead of fine-tuning the whole base pre-trained model for different NLP tasks (He et al., 2021).

- In the training step, the Mean Teacher loss (Baek et al., 2021) was added to improve model’s over-fitting problem.

- We explored data augmentation a method based on Monte Carlo (MC) dropout (Gal and Ghahramani, 2016) which to enhance the performance in sentence-level Direct Assessment (DA) and Multidimensional Quality Metrics (MQM) score task. During prediction, the dropout function is still enabled, and the prediction is performed for N times. The average value of the prediction is the final prediction value.

\textsuperscript{*} Indicates equal contribution.
\textsuperscript{1}https://wmt-qe-task.github.io/
\textsuperscript{2}https://github.com/Unbabel/COMET
• We used QE model’s predictor of the sentence-level quality prediction sub-task as a words’ features. extractor cosine similarity’s opposite value of target words’ vectors extracted from the predictor trained from sentence-level quality prediction sub-task between source and target as the explainable QE task’s token-level scores.

Our methods achieve impressive performance on both sentence- and word-level tasks. Specifically, we peak the top-1 on quality prediction sentence-level sub-task over Chinese-English language pair and word-level sub-task over English-Japanese language pair. We also win the first place in explainable QE task in Khmer-English and Pashto-English language pairs. We will describe the datasets and our methods for those tasks in section 2 and section 3. Section 4 presents details of our experimental setup and results. In section 5, a brief discussion and conclusion are presented.

2 Task & Data Set

2.1 Task Description

Task 1

The quality prediction task follows the trend of the previous years in comprising a sentence-level sub-task where the goal is to predict the quality score for each source-target sentence pair and a word-level sub-task where the goal is to predict the translation errors, assigning OK/BAD tags to each word of the target. Both sub-tasks include annotations derived in two different ways, depending on the language pair: direct assessment (DA), following the trend of the previous years, and multidimensional quality metrics (MQM), introduced for the first time in the QE shared task. The sentence- and word-level sub-tasks use the same source-target sentences for each language pair.

Task 2

The explainable QE task proposes to address translation error identification as rationale extraction. Instead of training a dedicated word-level model, to infer translation errors as an explanation for sentence-level quality scores, a list of continuous token-level scores where the tokens with the highest scores are expected to correspond to translation errors should be calculated.

2.2 Data Set & Data Processing

Some information about the data set is as follow:

There are three language pairs annotated with MQM annotations for training/development/test set: English-Russian (En-Ru), English-German (En-De), Chinese-English (Zh-En) and the one language pair annotated with DA annotations for training/development/test set: English-Marathi (En-Mr).

The data set of these four language pairs contains 15k training data for En-Ru, 26k training data for En-De, 31k training data for Zh-En, 26k training data for En-Mr and 1k development data for each language pair.

There are seven language pairs annotated with DA annotations for training/development set: English-German (En-De), English-Chinese (En-Zh), Estonian-English (Et-En), Nepali-English (Ne-En), Romanian-English (Ro-En), Russian-English (Ru-En), Sinhala-English (Si-En), and four zero-shot language pairs annotated with DA annotations for test set: English-Czech (En-Cs), English-Japanese (En-Ja), Khmer-English (Km-En) and Pashto-English (Ps-En). The data set of these seven language pairs contains 9k training data and 1k development data.

The word-level sub-task data set consists of predicting word-level tags for the target side (to detect mistranslated or missing words). Each token is tagged as either OK or BAD. The OK/BAD tags are provided for each of the language pairs of the sentence-level task, and are derived from either MQM annotations (En-De, Zh-En and En-Ru) or post-edited sentences.

So for MQM language pairs, it is a few-shot task, and for DA language pairs, it is a zero-shot task. For training data of each language pair, sentence scores are linearly normalized from 0 to 1, and can be restored to the original value, so a multilingual sentence-level QE model can be trained for all language pairs.

3 Methodology

3.1 System

Task 1

Our quality estimator system follows the two-stage Predictor-Estimator architecture, which uses a language model encoder as predictor and using task-specific classifier or regressor as estimator (Chen et al., 2021). In our system, the predictor is a pre-trained cross-lingual XLM-RoBERTa model (f). For the sentence-level quality score prediction task, the estimator is a regressor ($\tilde{\sigma}_{\text{score}}$), and for
the word-level quality label prediction task, the estimator is a classifier (σ_class), as depicted in figure 1.

\[ H_{sA}, H_{tA} = f([S, T]), \]
\[ F_{sA}, F_{tA} = p(H_{sA}, H_{tA}), \]
\[ H_{tB}, H_{sB} = f([T, S]), \]
\[ F_{tB}, F_{sB} = p(H_{tB}, H_{sB}), \]
\[ V_{\text{score}} = \sigma_{\text{score}}(F_{sA}, F_{tA}, F_{sB}, F_{tB}), \]
\[ V_{\text{class}} = \sigma_{\text{class}}([H_{sB}, H_{tB}]), \]
\[ V_{w-\text{score}} = \text{softmax}(V_{\text{class}}[0]) \]

Where \( V_{\text{score}} \) is output of the sentence-level estimator and \( V_{\text{class}} \) is logits of the word-level estimator.

**Task 2**

We use the sentence-level QE model’s predictor from task 1 as a sentence word embedding feature extractor. Similarity is used as the possibility of word translation (Yang et al., 2022). If a word in target is highly similar to a word in source, the word translation is correct. Otherwise, the word translation is incorrect. The higher the similarity, the higher the probability of correct translation, and vice versa.

We extracted the source and target sentence embedding features by word and calculated the cosine similarity between each word feature in the target and each word feature in the source. The maximum similarity between each word in the target and the source is used as the score of the word translation quality. We used the inverse of the quality score of each word in the target as the translation error risk value, so each target sentence can obtain a word error risk value list, in which a higher score indicates a higher probability of incorrect translation.

**3.2 Model Pre-training**

**Cross-lingual Language Model**

As XLM-RoBERTa, a multilingual model that can override the QE tasks’ language pairs, does a good job with language tasks, it was chosen as the predictor. Cross-lingual language model pre-training is outstanding in low-resource training data. We add [CLS] between the tokens of the source text and the tokens of the target text and input the combined tokens to the XLM-RoBERTa model for masked language modeling (MLM) task pre-training (Devlin et al., 2018) to enhance the model’s ability to understand words and sentences between languages. We sampled randomly 15% of the sub-word tokens from the text streams, replaced them by a [MASK] token in 80% probability, by a
random token in 10% chance, and we kepted them unchanged in 10% chance.

**Token Shuffling Pre-training**

We randomized the sequence of input tokens and let the cross-lingual language model predict the sequence number of each token. This pre-training task has an obvious positive effect on word-level QE sub-task. Because the model has never done a position prediction task, the training task is divided into two stages for the sake of training stability. In stage one, 50% of the tokens are selected and shuffled, and in stage two, all the tokens are shuffled.

### 3.3 Bottleneck Adapter Layer

The provided training set is relatively small, making the model to be easily over-fitted if all weights are updated. Therefore, we decided to integrate the Bottleneck Adapter Layers (BAL) (Wang et al., 2020) while keeping parameters of the original Transformer fixed (Yang et al., 2020). The bottle with a "thick" neck could further improve the performance without seriously sacrificing training efficiency. By increasing the parameter size of BALs, the performance also increased linearly, finally reaching the peak of 104% of the baseline performance with the neck having twice the hidden size.

### 3.4 Model Training

#### Mean Teacher Loss

Mean teacher is a method that uses consistency regularization. As shown in figure 2, the process is as follows:

1) Copy the predictor as a teacher model and the original model as a student model.

2) At the training step, apply two random augmentations $\eta$ and $\eta'$ on the same mini-batch tokens’ embedding features.

3) Input the former data $(Input_{embedding} + \eta)$ to the student model and the latter data $(Input_{embedding} + \eta')$ to the teacher model.

4) Calculate the MSE loss on their outputs.

5) Use the MSE loss to update the $t$ th iter’s parameters of the student model $P_{stu}[t]$.

6) Use the exponential moving average (EMA) method to update the $t$ th iter’s parameters of the teacher model $P_{tea}[t]$ as shown in equation 2.

$$P_{tea}[t] = \alpha \times P_{tea}[t-1] + (1-\alpha) \times P_{stu}[t] \quad (2)$$

Where, $\alpha$ is a hyperparameter (0.95 in this paper).

#### MLM Task Loss

To further improve the language understanding capability of the model, we add MLM task loss into the training. We find adding MLM training task during the training of sentence-level and word-level QE models for multi-task training can improve the model performance.

Total loss for the CrossQE system to update model parameters is shown as equation 3:

$$Loss = \alpha_1 \times [Loss_s | Loss_w] + \alpha_2 \times Loss_{MT} + \alpha_3 \times Loss_{MLM} \quad (3)$$

Where, $\alpha_1$, $\alpha_2$ and $\alpha_3$ are hyperparameters, $[Loss_s | Loss_w]$ is the sentence-level or word-level sub-task training loss, $Loss_{MT}$ is the mean teacher loss and $Loss_{MLM}$ is the MLM task loss.

### 4 Experiments & Results

#### 4.1 Model Settings

We followed the model settings of COMET (Rei et al., 2022) to fine-tune our QE model based on the XLM-RoBERTa large model 3 with a classification/regression head on a single V100 GPU. The XLM-RoBERTa large model pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages is a multilingual version of RoBERTa which is a transformers model pre-trained on a large corpus in a self-supervised fashion. It has approximately 550M parameters and 24 hidden encode layers. The training batch size is set to 4, the gradient accumulation number is set to 4 and it took about 2 hours for the model to converge in the training step. The XLM-RoBERTa large model has been pre-trained on the WMT 2021 news translation shared task’s parallel corpora 4 by model pre-training methods described in the section 3.2.

---

3. https://huggingface.co/xlm-roberta-large

Table 1: Results of the task 1 sentence-level’s spearman coefficient performance on development set over ten language pairs.

<table>
<thead>
<tr>
<th>model</th>
<th>En-Ru</th>
<th>En-De</th>
<th>Zh-En</th>
<th>En-Mr</th>
<th>En-Zh</th>
<th>Et-En</th>
<th>Ne-En</th>
<th>Ro-En</th>
<th>Ru-En</th>
<th>Si-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>0.3852</td>
<td>0.4436</td>
<td>0.3148</td>
<td>0.5123</td>
<td>0.2437</td>
<td>0.4635</td>
<td>0.5379</td>
<td>0.3572</td>
<td>0.4699</td>
<td>0.6109</td>
</tr>
<tr>
<td>M-Cross</td>
<td>0.4403</td>
<td>0.4807</td>
<td>0.3796</td>
<td>0.5419</td>
<td>0.2911</td>
<td>0.4827</td>
<td>0.5744</td>
<td>0.3899</td>
<td>0.4712</td>
<td>0.6358</td>
</tr>
<tr>
<td>M-Adapter</td>
<td>0.4487</td>
<td>0.4926</td>
<td>0.3815</td>
<td>0.5547</td>
<td>0.2938</td>
<td>0.4913</td>
<td>0.5899</td>
<td>0.4003</td>
<td>0.4962</td>
<td>0.6471</td>
</tr>
<tr>
<td>M-MT</td>
<td>0.4531</td>
<td>0.4917</td>
<td>0.3827</td>
<td>0.5681</td>
<td>0.3094</td>
<td>0.5083</td>
<td>0.6092</td>
<td>0.4090</td>
<td>0.5101</td>
<td>0.6566</td>
</tr>
<tr>
<td>M-MLM</td>
<td>0.4599</td>
<td>0.4928</td>
<td>0.3812</td>
<td>0.5679</td>
<td>0.3008</td>
<td>0.5101</td>
<td>0.6044</td>
<td>0.4182</td>
<td>0.5062</td>
<td>0.6653</td>
</tr>
<tr>
<td>M-Final</td>
<td>0.4730</td>
<td>0.5228</td>
<td>0.4002</td>
<td>0.5937</td>
<td>0.3247</td>
<td>0.5336</td>
<td>0.6217</td>
<td>0.4483</td>
<td>0.5211</td>
<td>0.6973</td>
</tr>
</tbody>
</table>

Table 2: Results of the task 1 word-level’s target words’ MCC performance on development set over ten language pairs.

<table>
<thead>
<tr>
<th>model</th>
<th>En-Ru</th>
<th>En-De</th>
<th>Zh-En</th>
<th>En-Mr</th>
<th>En-Zh</th>
<th>Et-En</th>
<th>Ne-En</th>
<th>Ro-En</th>
<th>Ru-En</th>
<th>Si-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>0.3182</td>
<td>0.2777</td>
<td>0.2643</td>
<td>0.3655</td>
<td>0.4007</td>
<td>0.2653</td>
<td>0.4432</td>
<td>0.3705</td>
<td>0.3642</td>
<td>0.4201</td>
</tr>
<tr>
<td>M-Adapter</td>
<td>0.3248</td>
<td>0.2796</td>
<td>0.2711</td>
<td>0.3681</td>
<td>0.4052</td>
<td>0.2714</td>
<td>0.4506</td>
<td>0.3832</td>
<td>0.3795</td>
<td>0.4588</td>
</tr>
<tr>
<td>M-MT</td>
<td>0.3274</td>
<td>0.3003</td>
<td>0.2807</td>
<td>0.3617</td>
<td>0.4201</td>
<td>0.2885</td>
<td>0.4494</td>
<td>0.3997</td>
<td>0.3814</td>
<td>0.4473</td>
</tr>
<tr>
<td>M-Final</td>
<td>0.3671</td>
<td>0.3112</td>
<td>0.2997</td>
<td>0.3872</td>
<td>0.4447</td>
<td>0.2963</td>
<td>0.4704</td>
<td>0.4041</td>
<td>0.3894</td>
<td>0.4960</td>
</tr>
</tbody>
</table>

4.2 Experiments of Sentence-level QE Task

In our experiment, we set $\alpha_1 = 1.0$, $\alpha_2 = 0.5$ and $\alpha_3 = 0.5$ (we also set $\alpha_1 = 1.0$, $\alpha_2 = 1.0$, $\alpha_3 = 1.0$ or $\alpha_1 = 0.5$, $\alpha_2 = 1.0$, $\alpha_3 = 0.5$ or $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, $\alpha_3 = 1.0$ or $\alpha_1 = 0.5$, $\alpha_2 = 1.0$, $\alpha_3 = 1.0$, but all of them can not get the best result). Our baseline model is the COMET’s open-source framework model with the self-pretrained XLM-RoBERTa model as predictor. The primary evaluation metric for the sentence-level sub-task of Task 1 is the spearman $r$ coefficient as show in Table 1.

Obviously, the performance of the baseline model is relatively poor. By leveraging Cross-lingual language model as predictor (M-Cross model), the model achieved much better performance. Adding the BAL adapter (M-Adapter model) into Cross-lingual language model, the effect is further improved. In the experiment, it is found that excessive training leads to reduced effectiveness on development set, while the addition of mean teacher loss (M-MT model) can significantly suppress the overfitting problem and further improve the model performance. Adding the MLM loss (M-MLM model) to the training process enhances the model performance to some degree. Finally, the MC dropout method is used to predict the QE sentence-level scores (M-Final model), which can improve the performance coefficient by at least 1%.

Table 3: Results of the task 1 sentence-level’s spearman coefficient performance on the test set over eight language pairs.

<table>
<thead>
<tr>
<th>Language</th>
<th>Spearman</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-Ru</td>
<td>0.4329</td>
</tr>
<tr>
<td>En-De</td>
<td>0.4939</td>
</tr>
<tr>
<td>Zh-En</td>
<td>0.3685</td>
</tr>
<tr>
<td>En-Mr</td>
<td>0.5672</td>
</tr>
<tr>
<td>En-Cs</td>
<td>0.6257</td>
</tr>
<tr>
<td>En-Ja</td>
<td>0.3409</td>
</tr>
<tr>
<td>Km-En</td>
<td>0.5087</td>
</tr>
<tr>
<td>Ps-En</td>
<td>0.6608</td>
</tr>
</tbody>
</table>

Finally, we committed our results of M-Final model on the test set. The performance of the system on the test set is shown in Table 3. For the zero-shot data, the system also has good performance. Specifically, we get the best performance on Zh-En language pair.

4.3 Experiments of Word-level QE Task

In our experiment, we set $\alpha_1 = 0.5$, $\alpha_2 = 1.0$ and $\alpha_3 = 1.0$ (we also set $\alpha_1 = 1.0$, $\alpha_2 = 1.0$, $\alpha_3 = 1.0$ or $\alpha_1 = 0.5$, $\alpha_2 = 1.0$, $\alpha_3 = 0.5$ or $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, $\alpha_3 = 1.0$ or $\alpha_1 = 0.5$, $\alpha_2 = 1.0$, $\alpha_3 = 1.0$, $\alpha_2 = 0.5$, $\alpha_3 = 0.5$, but all of them can not get the best result). Our baseline model is the cross-lingual language model that is used as predictor by the COMET’s open-source framework. The primary evaluation
metric for the word-level sub-task of Task 1 is the Matthews correlation coefficient (MCC) as shown in Table 1.

Compared with the baseline, the model has better performance after the BAL adapter is added (M-Adapter model). Also, the addition of mean teacher loss (M-MT model) can further improve the model performance. However, we found after adding the MLM loss to the training process (M-Final model), there were no significant improvements in performance.

Finally, we committed our results of M-Final model on the test set. The performance of the system on the test set is shown in Table 4. For the zero-shot data, the system also has good performance. Specifically, we get the best performance on En-Ja language pair.

### 4.4 Experiments of Explainable QE Task

As stated in the mission requirements, the participants are not allowed to supervise their models with any token-level or word-level labels or signals (whether they are from natural or synthetic data) in order to directly predict word-level errors. We just used the sentence-level quality prediction model’s predictor as the sentence word embedding feature extractor, and calculated the translation error risk value as stated in section 3.1.

Finally, we committed our results on the test set. The performance of the system on the test set is shown in the Table 5. We get the best performance on the Km-En and Ps-En language pairs.

### 5 Conclusion

This paper presents HW-TSC’s work called CrossQE on WMT 2022 QE shared task. CrossQE got the first place in four single projects. For

<table>
<thead>
<tr>
<th>Language</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-Ru</td>
<td>0.3425</td>
</tr>
<tr>
<td>En-De</td>
<td>0.2739</td>
</tr>
<tr>
<td>Zh-En</td>
<td>0.2457</td>
</tr>
<tr>
<td>En-Mr</td>
<td>0.3509</td>
</tr>
<tr>
<td>En-Cs</td>
<td>0.4239</td>
</tr>
<tr>
<td>En-Ja</td>
<td>0.2576</td>
</tr>
<tr>
<td>Km-En</td>
<td>0.3531</td>
</tr>
<tr>
<td>Ps-En</td>
<td>0.3576</td>
</tr>
</tbody>
</table>

Table 4: Results of the task 1 word-level’s target words’ MCC performance on the test set over eight language pairs.

<table>
<thead>
<tr>
<th>Language</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-Ru</td>
<td>0.3126</td>
</tr>
<tr>
<td>En-De</td>
<td>0.2517</td>
</tr>
<tr>
<td>Zh-En</td>
<td>0.2203</td>
</tr>
<tr>
<td>En-Mr</td>
<td>0.2800</td>
</tr>
<tr>
<td>En-Cs</td>
<td>0.5356</td>
</tr>
<tr>
<td>En-Ja</td>
<td>0.4617</td>
</tr>
<tr>
<td>Km-En</td>
<td>0.6863</td>
</tr>
<tr>
<td>Ps-En</td>
<td>0.7151</td>
</tr>
</tbody>
</table>

Table 5: Results for the task 2 target recall at top-K’s performance on the test set over eight language pairs.

the tasks 1, CrossQE employed the predictor-estimator framework as baseline. To further boost performance, we investigated the usage of pre-trained cross-lingual XLM-RoBERTa large language model as predictor, and added the bottleneck adapter layer into the predictor to mitigate overfitting issues. For both sentence- and word-level sub-task, we added mean teacher loss and MLM task loss into model training step, and added MC dropout at the inference step in sentence-level subtask. Those methods delivered a good performance in all language pairs, including zero-shot language pairs. For task 2, we used the sentence-level QE model’s predictor from task 1 as a sentence word embedding feature extractor, and used the inverse value of maximum similarity between each word in the target and the source as the word translation error risk value. In future, we will invest time and effort in studying the effect of involving additional translations into QE tasks, for example, how the additional translation quality will affect QE performance.

### References


Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco


