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

Shimin Tao[1*], Chang Su[1*], Miaomiao Ma[1], Hao Yang[1], Min Zhang[1],
Xiang Geng[2], Shujian Huang[2], Jiaxin Guo[1], Minghan Wang[1], Yinglu Li[1]
[1]Huawei Translation Services Center, China
[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 employes 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 [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 [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 Multidimentional 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.

[*] Indicates equal contribution.
[1]https://wmt-qe-task.github.io/
[2]https://github.com/Unbabel/COMET

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 646–652
Abu Dhabi, December 7–8, 2022. ©2022 Association for Computational Linguistics

- 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 ser: English-German (En-De), English-Chinese (En-Zh), Esthonian-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 zeor-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 languange 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 (σ_{score}), and for the word-level quality label prediction task, the estimator is a classifier (σ_{class}), as depicted in figure 1.

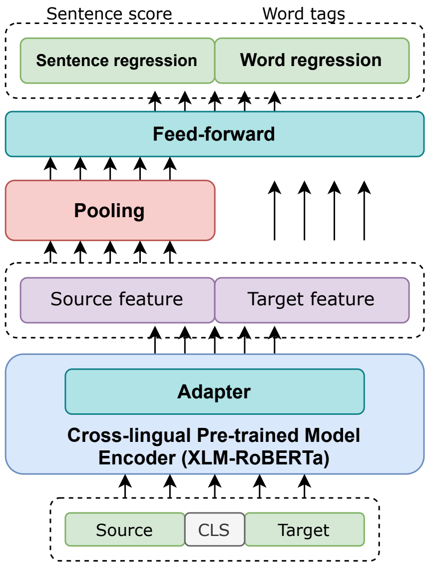

Figure 1: The architecture of CrossQE quality estimator system

Sentence-level

After the predictor obtaining tokens embedding features (H_s, H_t; where H_s for source embedding features and H_s for target embedding features), we use masked pooling p to calculate the entire source or target sentence feature vector. In the experiment, we put combination of [source, target] ($[S, T]$) and [target, source] ($[T, S]$) as the input data into the predictor, and get four types of sentence feature vectors ($F_{s_A}, F_{t_A}, F_{t_B}, F_{s_B}$). All the sentence feature vectors are combined to the estimator perform score prediction, and the performance is improved obviously.

Word-level

In the task, OK is set to 1 and BAD is set to 0, thus the word-level estimator becomes a binary classification model. To avoid overfitting, the OK label is set to 0.9, the BAD label is set to 0.1, and the index 0's value of outputs softmax logits is used as the word quality score ($V_{w-score}$). The mean squared error (MSE) loss is calculated on the outputs and labels and the word-level QE model is updated. In the prediction phase, if the output word score is greater than 0.5, it is considered as an OK label. Otherwise, it is considered as a BAD label.

The equation of task 1 is shown as equation 1:

$$\begin{aligned}
H_{s_A}, H_{t_A} &= f([S, T]), \\
F_{s_A}, F_{t_A} &= p(H_{s_A}, H_{t_A}), \\
H_{t_B}, H_{s_B} &= f([T, S]), \\
F_{t_B}, F_{s_B} &= p(H_{t_B}, H_{s_B}), \\
V_{score} &= \sigma_{score}([F_{s_A}, F_{t_A}, F_{s_B}, F_{t_B}]), \\
V_{class} &= \sigma_{class}([H_{s_B}, H_{t_B}]), \\
V_{w-score} &= softmax(V_{class})[0]
\end{aligned} \tag{1}$$

Where V_{score} is output of the sentence-level estimator and V_{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 keeped 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 η and η' 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, α is a hyperparameter (0.95 in this paper).

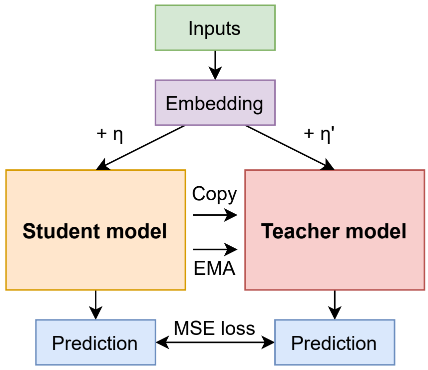

Figure 2: Mean teacher loss

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:

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

Where, α_1, α_2 and α_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 pretrained 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
[4] https://www.statmt.org/wmt21/translation-task.html

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

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

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

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

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 pre-trained 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 tearcher 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%.

Language	Spearman
En-Ru	0.4329
En-De	0.4939
Zh-En	0.3685
En-Mr	0.5672
En-Cs	0.6257
En-Ja	0.3409
Km-En	0.5087
Ps-En	0.6608

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

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 $\alpha_1 = 1.0$, $\alpha_2 = 0.5$, $\alpha_3 = 0.5$, 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

Language	MCC
En-Ru	0.3425
En-De	0.2739
Zh-En	0.2457
En-Mr	0.3509
En-Cs	0.4239
En-Ja	0.2576
Km-En	0.3531
Ps-En	0.3576

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

Language	Recall
En-Ru	0.3126
En-De	0.2517
Zh-En	0.2203
En-Mr	0.2800
En-Cs	0.5356
En-Ja	0.4617
Km-En	0.6863
Ps-En	0.7151

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

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 tearcher loss (M-MT model) can further improve the model pereformance. However, we found after adding the MLM loss to the training process (M-Final model), there were no significant improvement in pereformance.

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 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 over-fitting 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 sub-task. 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

Jeonghun Baek, Yusuke Matsui, and Kiyoharu Aizawa. 2021. What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3113–3122.

Yimeng Chen, Chang Su, Yingtao Zhang, Yuxia Wang, Xiang Geng, Hao Yang, Shimin Tao, Guo Jiaxin, Wang Minghan, Min Zhang, et al. 2021. Hw-tsc's participation at wmt 2021 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 890–896.

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

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si. 2021. On the effectiveness of adapter-based tuning for pretrained language model adaptation. *arXiv preprint arXiv:2106.03164*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

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*, 17:1–22.

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

Ricardo Rei, Ana C Farinha, José G.C. de Souza, Pedro G. Ramos, André F.T. Martins, Luisa Coheur, and Alon Lavie. 2022. Searching for COMETINHO: The little metric that could. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 61–70, Ghent, Belgium. European Association for Machine Translation.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Minghan Wang, Hao Yang, Hengchao Shang, Daimeng Wei, Jiaxin Guo, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, Yimeng Chen, et al. 2020. Hw-tsc's participation at wmt 2020 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1056–1061.

Hao Yang, Minghan Wang, Ning Xie, Ying Qin, and Yao Deng. 2020. Efficient transfer learning for quality estimation with bottleneck adapter layer. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 29–34.

Hao Yang, Min Zhang, Shimin Tao, Miaomiao Ma, Ying Qin, and Chang Su. 2022. TeacherSim: Cross-lingual machine translation evaluation with monolingual embedding as teacher. In *The 2nd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. To be publiushed.