HW-TSC 2024 Submission for the Quality Estimation Shared Task

Weiqiao Shan, Ming Zhu, Yuang Li, Mengyao Piao, Xiaofeng Zhao,
Chang Su, Min Zhang, Hao Yang, Yanfei Jiang
Huawei Translation Services Center, China
shanweiqiao96@gmail.com, {zhuming47, liyuang3, piaomengyao1, zhaoxiaofeng14,
suchang8, zhangmin186, yanghao30, jiangyanfei}@huawei.com

Abstract

Quality estimation (QE) is a crucial technique for evaluating the quality of machine translations without the need for reference translations. This paper focuses on Huawei Translation Services Center's (HW-TSC's) submission to the sentence-level QE shared task, named LLMs-enhanced-CrossQE. Our system builds upon the CrossQE architecture from our submission from last year, which consists of a multilingual base model and a task-specific downstream layer. The model input is a concatenation of the source and the translated sentences. To enhance performance, we fine-tuned and ensembled multiple base models, including XLM-R, InfoXLM, RemBERT, and CometKiwi. Specifically, we employed two pseudo-data generation methods: 1) a diverse pseudo-data generation method based on the corruption-based data augmentation technique introduced last year, and 2) a pseudo-data generation method that simulates machine translation errors using large language models (LLMs). Our results demonstrate that the system achieves outstanding performance on sentence-level QE test sets.

1 Introduction

Quality estimation (QE) aims to automatically assess machine translation outputs without requiring reference translations (Specia et al., 2018). We report the technical details of our approach to sentence-level quality prediction and fine-grained error span detection subtasks in the WMT 2024 QE shared task. Our team, Huawei Translation Services Center (HW-TSC), participated in direct assessment (DA) score in sentence-level quality prediction and the fine-grained error span detection tasks across all language pairs. Fine-tuning pre-trained language models, which provide rich semantic information, has become the standard approach for QE tasks (Rei et al., 2020). In this paper, we present LLMs-enhanced-CrossQE, HW-TSC's system for the sentence-level QE task, which leverages multiple pre-trained language models and data augmentation techniques. The key aspects of our system design are summarized as follows:

- **Model**: We employed our previous year's architecture, CrossQE (Tao et al., 2022), as the foundation. For every language pair, models were individually fine-tuned. Additionally, we used CometKiwi (Rei et al., 2022), a multilingual QE model, and fine-tuned it for single language pairs.

- **Data augmentation**: Based on the corruption-based data generation (CDG) method used last year (Li et al., 2023), we propose a diverse CDG (D-CDG) method. Specifically, we generate more varied corrupted translations by combining multiple error types. Additionally, we rewrite source sentences using large language models (LLMs) to create pseudo-sentences containing errors that closely resemble those produced by machine translation systems. Finally, we employ a reference-based QE model to generate pseudo scores.

- **Ensemble**: For each language pair, we ensemble eight fine-tuned models to achieve optimal performance. These checkpoints originated from four base models: XLM-R (Conneau et al., 2020), InfoXLM (Chi et al., 2021), RemBERT (Chung et al., 2020), and CometKiwi (Rei et al., 2022), and three training dataset configurations: original dataset, augmented dataset, and augmented dataset followed by the original dataset. The ensemble weight for each checkpoint was optimized with Optuna (Akiba et al., 2019). On average, eight checkpoints were used per language pair after optimization. Additionally, we experimented with a naive weight ensemble approach based on the method proposed by Yadav et al. (2024), but it did not yield significant improvements.

Proceedings of the Ninth Conference on Machine Translation (WMT), pages 535–540

Our system ranks first in the English-Tamil direction and second in several other directions in the direct assessment quality estimation task (Zerva et al., 2024). It significantly outperforms the baseline given by the competition organizers by a large margin. Additionally, we provide detailed results of each model with and without data augmentation in Table 3. To analyze the importance of each model in the ensemble, we present the ensemble weights in Figure 2 and 1. It is worth noting that the models fine-tuned with the proposed data augmentation technique were assigned higher weights in the ensemble.

2 Background

2.1 Task Description [1]

Sentence-level QE with direct assessment (DA) anotations: The goal is to predict the quality score for each source-target sentence pair. The golden-truth quality scores were obtained from human translators who rated each translation from 0 to 100. The scores from three or four translators were normalized and averaged to get the final score. This year's QE shared task has four language pairs with DA quality scores: English-Hindi (en-hi), English-Tamil (en-ta), English-Telegu (en-te) and English-Gujarati (en-gu). All languages have just 7,000 training samples.

Fine-grained error span detection: Participants of this task need to identify the error span (start and end indices) and the error severity (major or minor).

2.2 Base Models

- **XLM-R** (Conneau et al., 2020): A transformer-based masked language model trained on a massive multilingual corpus with more than two terabytes of data.
- **InfoXLM** (Chi et al., 2021): A cross-lingual pre-trained model that leverages multilingual masked language modeling, translation language modeling, and cross-lingual contrast learning.
- **RemBERT** (Chung et al., 2020): A rebalanced mBERT model with factorization of the embedding layers. The input embeddings are smaller and kept for fine-tuning, while the output embeddings are larger and discarded after pre-training.
- **CometKiwi** (Rei et al., 2022): A multilingual reference-free QE model that uses a regression approach and is built on top of InfoXLM. It has been trained on direct assessments from WMT17 to WMT20 and the MLQE-PE corpus.

3 Method

3.1 Model Architecture

3.1.1 Task1: Sentence-level QE with direct assessment (DA)

As shown in Equation 1 and 2, the embeddings of source sentence $\mathbf{s}$ and translated sentence $\mathbf{t}$ are concatenated in both orders $[\mathbf{s}, \mathbf{t}]$ and $[\mathbf{t}, \mathbf{s}]$ to form the input of pre-trained model f_{base}. The output token-level embedding sequences are processed by an average pooling layer to obtain vector representations $\mathbf{h}_{s1}$ and $\mathbf{h}_{t1}$ for source and translation respectively. These feature vectors are enhanced by taking their absolute difference and element-wise multiplication, as shown in Equation 3 and 4. Finally, all feature vectors are concatenated and fed into a regression head that predicts the final score y (Equation 5). This architecture enables information exchange between source and translated sentences at an early stage of the network and has proven to be significantly more effective than combining cross-lingual information after the pre-trained model.

$$\mathbf{h}_{s1}, \mathbf{h}_{t1} = f_{base}([\mathbf{s}, \mathbf{t}]) \quad (1)$$

$$\mathbf{h}_{t2}, \mathbf{h}_{s2} = f_{base}([\mathbf{t}, \mathbf{s}]) \quad (2)$$

$$\mathbf{f}_1 = [\mathbf{h}_{s1}, \mathbf{h}_{t1}, |\mathbf{h}_{s1} - \mathbf{h}_{t1}|, \mathbf{h}_{s1} \odot \mathbf{h}_{t1}] \quad (3)$$

$$\mathbf{f}_2 = [\mathbf{h}_{s2}, \mathbf{h}_{t2}, |\mathbf{h}_{s2} - \mathbf{h}_{t2}|, \mathbf{h}_{s2} \odot \mathbf{h}_{t2}] \quad (4)$$

$$y = f_{score}([\mathbf{f}_1, \mathbf{f}_2]) \quad (5)$$

3.1.2 Task2: Error span detection

For this task, we speculate that the understanding ability of large models may be helpful to the task, so we use the TowerInstruct-7B-v0.2 (Alves et al., 2024) model and the GPT-4o-mini (Islam and Moushi, 2024) model to cope with this task.

3.2 Data Augmentation

In this year's QE shared task, we adapted two data augmentation methods. 1) Text Editing, we implemented a D-CDG method based on the CDG proposed last year (Li et al., 2023), in which we constructed more diverse translation error data by

[1] https://wmt-qe-task.github.io/

Method	Description
Deletion	A random word in the translation was deleted.
Insertion	A random word in the translation was selected and inserted in a random position.
Substitution	A random word was replaced with another word in the translation.

Table 1: Three available text editing methods.

Method
You are a Gujarati to English machine translation system. I will give you a correct parallel data pair, rewrite the target language (English) sentence with mistakes that you may have made while doing the translation, including but not limited to incorrect words, adding extra words, Omitting crucial words, wrong numbers or dates, deleting words, exchanging the position of two words, wrong numbers, incorrect punctuation, incorrect capitalization, grammar errors. The correct parallel data is: "$SRC", "$TGT". please just output the target language with 20%, 35%, 50% mistake token of the target length.

Table 2: A prompt example for LLMs to generate pseudo QE training data based on a sample from the Gujarati to English QE training set, $SRC and $TGT represent the source and target languages in the sample, respectively.

incorporating multiple text editing approaches. 2) LLMs-generated pseudo-data. We generated translation data with errors more similar to those produced by machine translation systems using GPT-4o-mini and constructed parallel data pairs containing translation errors through the machine translation system.

For text editing, we employed three methods proposed last year to generate translation errors: Deletion, Insertion, and Substitution. Notably, this year, we generated translation sentences with more diverse translation errors by combining these three text editing methods with a certain probability. Specifically, each time we performed a text edit, we modified the original text with equal probability by sampling a text editing method from a subset of the three available text editing methods. Additionally, we also created a version of pseudo-data by directly translating the source language into the target language and then back-translating it.

For LLM-generated pseudo-data, we constructed a prompt using the GPT-4o-mini to generate a modified source language sentence multiple times with different proportions, correlating with the number of tokens in the sentence(see Table 2). This approach yielded multiple modified source language sentences containing error tokens that closely resemble those generated by translation systems. These modified sentences were then translated into the target language using a translation system. Similar to the text editing method, we scored the pseudo-parallel translation pairs using a reference-based QE model [2] to create pseudo QE training data. It is worth noting that we constructed the scaling factor as the ratio between the corrupted translation score and the uncorrupted translation score ($\frac{f_{QE}(s,\hat{t},t)}{f_{QE}(s,t,t)}$), following the approach from last year.

4 Experiments

4.1 Experimental setups

Our system is built on top of the COMET package [3]. We fine-tuned four pre-trained models, namely XLM-R, InfoXLM, RemBERT and CometKiWi [4], on a single Nvidia Tesla V100 GPU with a batch size of 4, gradient accumulation of 8 and mean square error loss function. We stopped the training when there was no improvement in terms of Spearman correlation on the dev set for five test runs. For each language pair, the augmented dataset from text editing method, which contains more than ten times data than the original dataset, and the augmented dataset from LLMs, which contains about three times data than the original dataset, were all pre-generated instead of generated on-the-fly to improve training efficiency. Following last year's conclusion that the pseudo-data is more effective compared with the original data, we fine-tuned four base models by pseudo-data directly. The training step took around 10 hours with the augmented

[2] https://huggingface.co/Unbabel/wmt22-comet-da

[3] https://github.com/Unbabel/COMET

[4] https://huggingface.co/Unbabel/wmt22-cometkiwi-da

Method	en-hi	en-ta	en-te	en-gu	Avg.
XLM-R	0.616	0.663	0.434	0.643	0.589
+ aug (D-CDG)	0.614 (-.002)	0.675 (+.012)	0.449 (+.015)	0.657 (+.014)	0.599 (+.010)
+ aug (LLMs)	0.469 (-.147)	0.617 (-.046)	0.412 (-.022)	0.603 (-.040)	0.525 (-.064)
InfoXLM	0.595	0.670	0.443	0.664	0.593
+ aug (D-CDG)	0.608 (+.013)	0.657 (-.013)	0.465 (+.022)	0.671 (+.007)	0.600 (+.007)
+ aug (LLMs)	0.478 (-.117)	0.614 (-.056)	0.418 (-.025)	0.629 (-.035)	0.535 (-.058)
RemBERT	0.606	0.671	0.431	0.688	0.599
+ aug (D-CDG)	0.604 (-.002)	0.672 (+.001)	0.432 (+.001)	0.667 (-.021)	0.594 (-.005)
+ aug (LLMs)	0.458 (-.148)	0.606 (-.065)	0.413 (-.018)	0.617 (-.071)	0.524 (-.075)
CometKiwi	0.590	0.685	0.451	0.691	0.604
+ aug (D-CDG)	0.594 (+.004)	0.683 (-.002)	0.465 (+.014)	0.696 (+.005)	0.610 (+.006)
+ aug (LLMs)	0.475 (-.115)	0.630 (-.055)	0.420 (-.031)	0.662 (-.029)	0.547 (-.057)
Ensemble (D-CDG)	0.652 \ **0.719**	0.716 \ 0.675	0.483 \ **0.482**	0.717 \ 0.678	0.642 \ 0.637
Ensemble (LLMs)	**-**	0.712 \ **0.683**	0.48 \ 0.474	0.714 \ **0.686**	0.635 \ 0.614
Ensemble (submit)	- \ **0.719**	- \ **0.683**	- \ **0.482**	- \ **0.686**	- \ **0.643**

Table 3: Results for sentence-level QE in terms of **Spearman** correlation. We report the performance of using D-CDG and LLM-generated pseudo-data as a data augmentation approach(aug). Except for the last three rows which show the results on the dev \ test set, other results were based on the dev set.

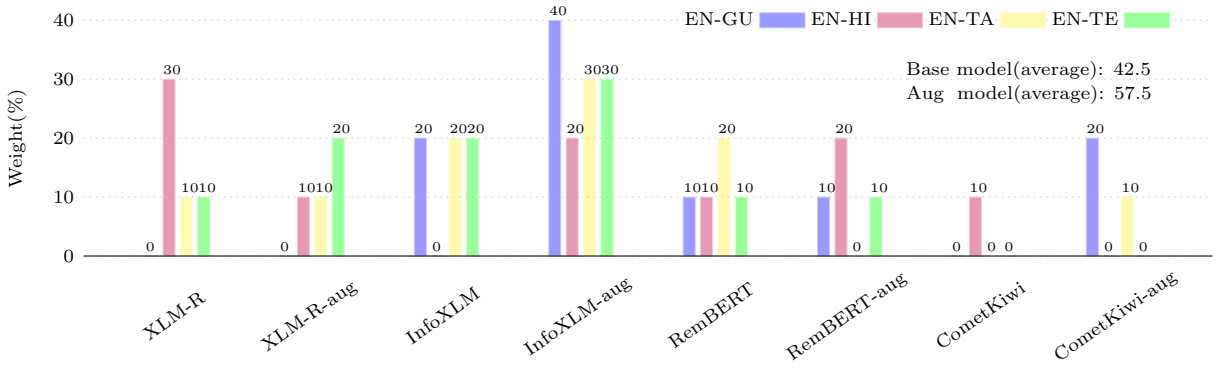

Figure 1: The ensemble weights for each base model.

dataset.

With four base models and two data augmentation approaches, we obtained eight checkpoints for each language pair. We ensembled these checkpoints by taking the weighted average of the predicted scores. The weights were optimized using Optuna, an automatic hyperparameter search framework. We used the Spearman correlation as the optimization objective, setting the step size to 0.05, and conducted 1000 trials on the dev set.

4.2 Results

4.2.1 Task1

The results of sentence-level QE in terms of Spearman correlation are shown in Table 3. Without data augmentation, CometKiwi has the best average correlation of 0.604, while XLM-R, InfoXLM, and RemBERT are close behind with around 0.590.

For the two data augmentation methods, we found that the D-CDG approach led to improvements across nearly all languages and models, as shown in Table 3. Additionally, this approach outperformed the original CDG method[5]. This suggests that rewriting a sentence by combining multiple diverse text editing methods within the same sentence is more effective than using only a single text editing method. Instead, for the pseudo-data generated by LLMs, we did not observe a positive effect on the dev set in all language directions, as shown in Table 3.

Furthermore, in the model ensemble, we observed that models with the D-CDG approach played a more important role. Specifically, the In-

[5]Reported in Table 1 of last year's paper

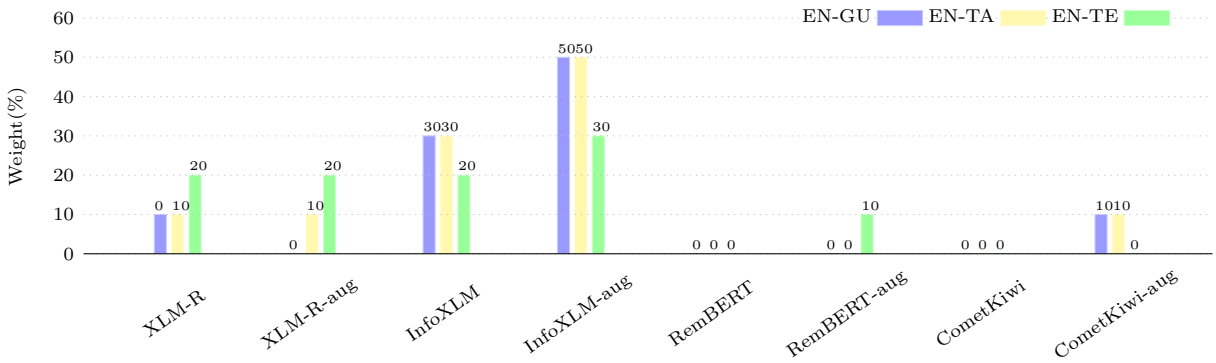

Figure 2: The ensemble weights for different training dataset configurations. 'w/o aug' and '+ aug' mean using the original or augmented dataset respectively. '+ aug & finetune' means training on the augmented dataset and then finetuning on the original one.

Method	**en-de**			**en-hi**			**en-es**		
	F1	recall	precison	F1	recall	precison	F1	recall	precison
Tower-Instruct-7B	0.178	0.181	0.175	0.015	0.008	0.300	0.118	0.082	0.209
GPT-4o-mini	0.119	0.315	0.073	0.361	0.398	0.331	0.146	0.249	0.103
Baseline **(test set)**	**0.192**	-	-	**0.481**	-	-	**0.161**	-	-
Ensemble **(test set)**	0.178	0.181	0.175	0.361	0.398	0.331	0.141	0.227	0.102

Table 4: Results for error span detection in terms of F1 score.

foXLM model with D-CDG was assigned a larger weight across all languages, as shown in Figure 1. We also noticed that the assignment of higher weights to models with D-CDG in the ensemble correlated with the base model's overall importance, if a base model received substantial attention, the corresponding model with D-CDG also tended to receive more weight.

Notably, models with LLM-generated pseudo-data were not assigned higher weights in the ensemble 2. However, in the test set, models with LLM-generated pseudo-data achieved better Spearman correlation scores in two languages(en-ta and en-gu). This may be attributed to the fact that LLMs generate more diverse pseudo-data, thereby enhancing the ensemble model's generalization ability. On the other hand, there may be a large gap between the dev set and the test set, the model with text editing data is overfitted to the dev set, while the models with LLMs pseudo-data introduce some regularization ability, which makes the ensemble model achieve better results on some languages.

This may be attributed to the fact that LLMs generate more diverse pseudo-data, thereby enhancing the ensemble model's generalization ability. Additionally, the discrepancy between the development set and the test set might have caused overfitting in models trained with text editing data. In contrast, models incorporating LLM-generated pseudo-data introduced a regularization effect, enabling the ensemble model to achieve better results in certain languages.

4.2.2 Task2

The results for error span detection are displayed in Table 4. In the Table, we can see that the method of using the large language model alone to detect the error segment is lower than the baseline based on cometkiwi, but it is not far from it. In addition, we can see that the method based on GPT-4o-mini is much higher than the method without LLMs in recall. That's enough to see the potential of the large language models, if human preferences can be injected for fine tuning, there is a good chance that large language models will outperform cometkiwi-based methods.

5 Conclusion

This paper mainly presents HW-TSC's sentence-level QE system called LLMs-enhanced-CrossQE. Using our previous year's model CrossQE as the foundation, we conducted comprehensive experiments with various pre-trained models. To further enhance the robustness of all language pairs and provide various checkpoints for model ensem-

ble, we introduced a diverse pseudo-data generation method based on the corruption-based data augmentation technique proposed last year. Our system demonstrates strong performance across all language pairs with DA annotations in the sentence-level QE task. In the future, we plan to explore the use of LLMs to generate more diverse QE pseudo-data using more effective in-context learning techniques, such as chain-of-thought (CoT) prompting, or by transferring knowledge from LLMs to QE models through direct utilization of LLM parameters. Additionally, this paper presents only a brief investigation of the error span detection task. Therefore, we plan to further explore word-level and document-level QE tasks, which can improve the interpretability of QE and hold significant promise in the era of LLMs.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proc. ACM SIGKDD*, pages 2623–2631.

Duarte M Alves, José Pombal, Nuno M Guerreiro, Pedro H Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, et al. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, He-Yan Huang, and Ming Zhou. 2021. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *Proc. NAACL*, pages 3576–3588.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2020. Rethinking embedding coupling in pre-trained language models. *arXiv preprint arXiv:2010.12821*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proc. ACL*, pages 8440–8451.

Raisa Islam and Owana Marzia Moushi. 2024. Gpt-4o: The cutting-edge advancement in multimodal llm. *Authorea Preprints*.

Yuang Li, Chang Su, Ming Zhu, Mengyao Piao, Xinglin Lyu, Min Zhang, and Hao Yang. 2023. Hw-tsc 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 835–840.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. In *Proc. EMNLP*, pages 2685–2702.

Ricardo Rei, Marcos Treviso, Nuno M Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José GC De Souza, Taisiya Glushkova, Duarte Alves, Luísa Coheur, et al. 2022. Cometkiwi: Ist-unbabel 2022 submission for the quality estimation shared task. In *Proc. WMT*, pages 634–645.

Lucia Specia, Carolina Scarton, Gustavo Henrique Paetzold, and Graeme Hirst. 2018. *Quality estimation for machine translation*, volume 11. Springer.

Shimin Tao, Su Chang, Ma Miaomiao, Hao Yang, Xiang Geng, Shujian Huang, Min Zhang, Jiaxin Guo, Minghan Wang, and Yinglu Li. 2022. Crossqe: Hw-tsc 2022 submission for the quality estimation shared task. In *Proc. WMT*, pages 646–652.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36.

Chrysoula Zerva, Frederic Blain, Jos'e G. C. de Souza, Diptesh Kanojia, Sourabh Deoghare, Nuno M. Guerreiro, Giuseppe Attanasio, Ricardo Rei, Constantin Orasan, Matteo Negri, Marco Turchi, Rajen Chatterjee, Pushpak Bhattacharyya, Markus Freitag, and Andr'e Martins. 2024. Findings of the quality estimation shared task at wmt 2024. are llms closing the gap in qe? In *Proceedings of the Ninth Conference on Machine Translation*. Association for Computational Linguistics.