IOL Research's Submission for WMT 2023 Quality Estimation Shared Task

Zeyu Yan, Wenbo Zhang, Qiaobo Deng,
Hongbao Mao, Jie Cai, Zhengyu He
Transn IOL Research, Wuhan, China
{zeyu.yan,albert01.zhang,qiaobo.deng,hubben.mao,jay.cai,steven.he}@transn.com

Abstract

This paper presents the submissions of IOL Research in WMT 2023 quality estimation shared task. We participate in task 1 Quality Estimation on both sentence and word levels, which predicts sentence quality score and word quality tags. Our system is a cross-lingual and multitask model for both sentence and word levels. We utilize several multilingual Pretrained Language Models (PLMs) as backbones and build task modules on them to achieve better predictions. A regression module on PLM is used to predict sentence level score and word tagging layer is used to classify the tag of each word in the translation based on the encoded representations from PLM. Each PLM is pretrained on quality estimation and metrics data from the previous WMT tasks before finetuning on training data this year. Furthermore, we integrate predictions from different models for better performance while the weights of each model are automatically searched and optimized by performance on Dev set. Our method achieves competitive results.

1 Introduction

Quality Estimation (QE) is the task of predicting the quality of a target machine translation without using reference texts or human inputs (Specia et al., 2018). Since machine translation is in high demand nowadays, the development of QE system becomes crucial for the broad application of machine translation. In WMT 2023 Quality Estimation shared task, there are two tasks: quality estimation and fine-grained error span detection. This paper describes our submission to task 1 quality estimation in both sentence and word levels in detail.

Considering the powerful capability and widely used in previous QE tasks of pretrained language models (PLMs) (Zerva et al., 2022; Specia et al., 2021), our method utilizes different multilingual PLMs to encode *source-translation* sentence pairs and predict sentence-level scores or word-level tags. Such PLMs are pretrained on various languages which could show incredible ability when trained QE models are transferred to unseen language pairs. Meanwhile, extra task modules are added to PLMs to boost the interaction between source and translation sentences to make better predictions. Also, it is common that using other task data similar to QE can further improve the performance of QE. According to the results from previous years' QE tasks, we use the data from QE and Metrics tasks from previous years' WMT tasks, as well as Automatic Post-Editing (APE) data, to pretrain PLMs before training on data of this year.

Moreover, ensemble methods of different models are explored in sentence and word level tasks. For sentence level, we sum scores with weights from different models which are filtered by the performance on the Dev set. As for word level, we use voting or weighted sum of tag probabilities to get the final predicted tags. Taking zero-shot language pairs into account, we choose the best model evaluated on other language pairs to test if they can generalize to unseen language pairs.

2 Quality Estimation Task

2.1 Task description

WMT 2023 Quality Estimation task 1 contains two tasks. The sentence level task aims at predicting a quality score for *translation* and the word level task is to classify a quality tag for each word in *translation*. Both tasks have zero-shot language pairs to test the generalization ability of QE models and use the same *source-translation* pairs for each language pair.

Sentence level There are two types of quality scores. One is the Direct Assessments (DA) score which is given by human annotators for each *source-translation* pair. The other is the Multi-dimensional Quality Metrics (MQM) score which is defined and computed under MQM methodology.

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 863–871
December 6–7, 2023.

train_sent	train_word	train_mtl
224195	263184	105992

Table 1: The statistics of train data

	Dev	Test
En-De	511	1897
Zh-En	505	1677
En-Mr	1000	1086
En-Gu	1000	1075
En-Hi	1000	1074
En-Ta	1000	1075
En-Te	1000	1075
He-En	-	1182
En-Fa	-	1000

Table 2: The statistics of dev and test data

A regression model is always employed to predict quality scores.

Word level The tags of words in *translation* are annotated by human annotators according to the MQM or DA annotations. This task requires predicting an OK or BAD for each word in *translation* given *source-translation* pairs. HTER (Specia and Farzindar, 2010)-like scores for translations can be collected by calculating the ratio of 'BAD' tags in tag sequence of *translations*. For example, given a tag sequence "OK OK BAD BAD OK", an HTER-like score is deduced by computing 2/5=0.4.

Data QE task provides official train and dev datasets gathered from competitions of previous years and the statistics are shown in Table 1 and Table 2. On account of the task similarity to QE, we also collect the MQM data (Freitag et al., 2021a,b) from previous WMT Metrics tasks[1] and APE data from QT21 (Specia et al., 2017) and APE-QUEST (Depraetere et al., 2020) to do further pretraining. We calculate HTER-like score for each *source-translation* pair in APE data for the purpose of merging with those of DA and MQM.

3 Method

3.1 Model architecture

We design distinct task modules on top of encoders for regression on sentence level and sequence tagging on word level. Source and translation texts are concatenated and input into the encoder and then task modules to get scores or tags. Our model architecture is illustrated in Fig.1.

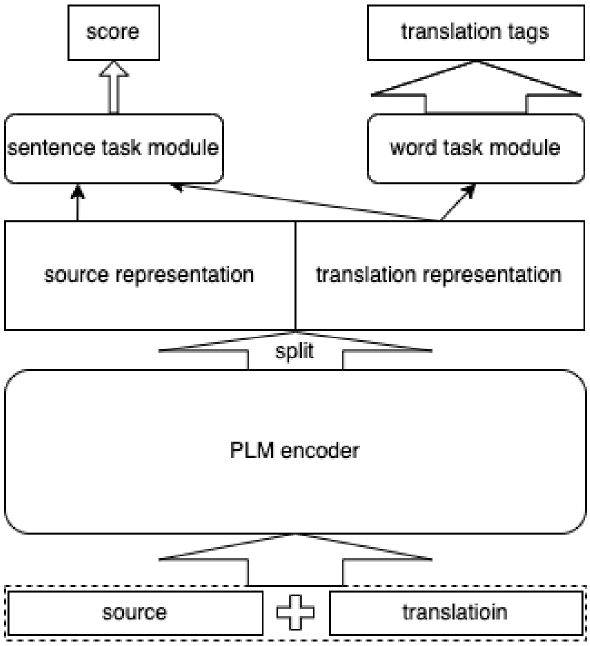

Figure 1: Model architecture with task modules for sentence-level scoring and word-level tagging

Sentence regression module Inspired by ESIM (Chen et al., 2017) and RE2 (Yang et al., 2019), the cross attention between *source* and *translation* reflects the similarity between words in different languages. Also considering that different layers in a transformer (Vaswani et al., 2017) based PLM catch different granularities of features of *source* and *translation* (Jawahar et al., 2019), we determine to combine these two kinds of methods to strengthen the representations of *source* and *translation*. In detail, for *source* s and *translation* t respectively, mixed layer-wise representations s_{mix} and t_{mix} from a PLM with L layers are computed in Eq. 1~Eq. 4.

$$s^l = mean\_pooling([s_1^l, s_2^l, ..., s_m^l]), \quad (1)$$

$$t^l = mean\_pooling([t_1^l, t_2^l, ..., t_n^l]), \quad (2)$$

$$s_{mix} = \sum_{l=1}^{L} w_s^l * s^l, \ where \sum_{l=1}^{L} w_s^l = 1 \quad (3)$$

$$t_{mix} = \sum_{l=1}^{L} w_t^l * t^l, \ where \sum_{l=1}^{L} w_t^l = 1 \quad (4)$$

Then cross attention outputs s_{ca} and t_{ca} from the last layer of PLM are calculated to get token level

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

interactions between *source* and *translation* as shown in Eq. 5 ~Eq. 9.

$$e_{ij} = s_i^T t_j \tag{5}$$

$$s_i^{ca} = \sum_{j=1}^{n} \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})} t_j,\ \forall i \in [1, 2, ..., m] \tag{6}$$

$$t_j^{ca} = \sum_{i=1}^{m} \frac{exp(e_{ij})}{\sum_{k=1}^{m} exp(e_{kj})} s_i,\ \forall j \in [1, 2, ..., n] \tag{7}$$

$$s_{ca} = mean\_pooling([s_1^{ca},\ s_2^{ca},\ ...,\ s_m^{ca}]) \tag{8}$$

$$t_{ca} = mean\_pooling([t_1^{ca},\ t_2^{ca},\ ...,\ t_n^{ca}]) \tag{9}$$

Next, features of *source* and *translation* are fused separately to transform into a combined representation through feedforward network (FFN) layer by Eq. 10 and Eq. 11.

$$s_{comb} = FFN([s_{ca}; s_{mix}; |s_{ca} - s_{mix}|; s_{ca} * s_{mix}]) \tag{10}$$

$$t_{comb} = FFN([t_{ca}; t_{mix}; |t_{ca} - t_{mix}|; t_{ca} * t_{mix}]) \tag{11}$$

Finally, the sentence-level score is obtained by another FFN layer in Eq. 12.

$$score = FFN([s_{comb}; t_{comb}]) \tag{12}$$

Word tagging module We choose two distinct modules to generate tags for words after encoded by PLM. A Bidirectional-LSTM (Hochreiter and Schmidhuber, 1997) (BiLSTM) layer is added to enhance the interaction between the representations of *source* and *translation*, and a FFN layer on it to predict tags in *translation*. Another kind of module only adopts a FFN layer to generate tag predictions to avoid overfitting on training data.

Multitask combination In order to boost the individual performance of sentence level and word level models, we propose a multitask training approach. Both the regression module and tagging module are added to the encoder which predicts the sentence score like DA or MQM and word tags simultaneously. For language pairs that have no DA or MQM data but only word tags, we take HTER scores as sentence scores. We train word-level models by optimizing the prediction of HTER scores and word tags simultaneously. To not damage the potentiality of tagging module, some simple regression modules are used when doing multitask training, including

$$score = FFN(mean\_pooling(\mathbf{t}_{[1:n]})) \tag{13}$$

and

$$score = FFN([\bar{s}; \bar{t}; |\bar{s} - \bar{t}|; \bar{s} * \bar{t}]) \tag{14}$$

where $\mathbf{t}_{[1:n]}$ is the list of word representations of *translation* from tagging module, and $\bar{s}$ and $\bar{t}$ are the mean representations of words' representations of *source* and *translation* from the encoder.

Loss The losses for score regression, word tagging and multitask training are described as follows:

$$\mathcal{L}_{sent} = (score_{pred} - score_{true})^2 \tag{15}$$

$$\mathcal{L}_{word} = -\frac{1}{n} \sum_{i=1}^{n} \log p(y_i) \tag{16}$$

$$\mathcal{L}_{multitask} = \mathcal{L}_{sent} + \mathcal{L}_{word} \tag{17}$$

where $p(y_i)$ is the probability of OK/BAD tag from the model.

3.2 Score refinement

According to the similar definitions and score intervals of DA and MQM, we transform the score s out of [-1, 1] as close to [-1,1] as possible while keeping the Spearman correlation coefficient unchanged using Eq. 18 to lessen the need for predicting extreme values during training.

$$s' = \begin{cases} \text{(s+1)*0.1-1,} & \text{s} < -1 \\ \text{s,} & \text{-1} \leq \text{s} \leq 1 \\ \text{(s-1)*0.1+1,} & \text{s} > 1 \end{cases} \tag{18}$$

3.3 Encoder selection

QE requires texts from different languages as input, so we take multilingual PLMs as encoders which are pretrained on colossal multilingual corpus. The following PLMs are selected as encoders: XLM-Roberta-Large (Conneau et al., 2020)[2], RemBert (Chung et al., 2021)[3], InfoXLM-Large (Chi et al., 2021)[4] and mDeBERTa (He et al., 2021)[5]. Each PLM is combined with different task modules for training.

[2]https://huggingface.co/xlm-roberta-large
[3]https://huggingface.co/google/rembert
[4]https://huggingface.co/microsoft/infoxlm-large
[5]https://huggingface.co/microsoft/mdeberta-v3-base

3.4 Model Training

We first pretrain encoders with a simple regression head to do regression on WMT Metrics and HTER data while retaining the checkpoints of encoders with the best performance on Dev set. When using WMT Metrics data, we train two versions of models where one uses DA data only and the other uses a mix of DA and MQM data. Subsequently, we drop the regression head and then finetune the pretrained encoder with different task modules on multilingual QE data. In order to eliminate the possible side effect of position variation in *translation*, we swap the input order of *source* and *translation* as a comparison. We conduct single-task and multitask training for both sentence and word levels.

3.5 Ensemble methods

Sentence level For each language pair having training data, we randomly search weights for the weighted sum of the top 10 models in accordance with the Spearman correlation coefficient on Dev set. As for zero-shot language pairs, we pick the best two or three trained models from those language pairs having training data individually then predict and average the scores from them.

Word level We propose three strategies of tag prediction ensemble for each word. At first, for each language pair having training data, the top 10 models with the best Matthews Correlation Coefficient on Dev set are picked out. Therefore each word in *translation* has 10 predicted tags or 10 probability pairs of (OK, BAD) from different models. The final tag of one word is acquired in one of three ways:

1 if one of 10 tags is BAD, the final tag is BAD;

2 if one of 10 tags is OK, the final tag is OK;

3 if the weighted sum of probabilities of OK is larger than that of BAD, the final tag is OK, and vice versa.

When utilizing the third one, the weights of models are searched randomly as in sentence-level ensemble. As for zero-shot language pairs, we pick the best two trained models from those language pairs having training data individually and apply one of the above strategies to get final predictions.

4 Experiments

4.1 Settings

All our models are completed with PyTorch and transformers (Wolf et al., 2020)[6] and trained on NVIDIA GeForce RTX 3090 24G for the pretraining and finetuning described in 3.4. Models are trained with AdamW (Loshchilov and Hutter, 2017) with learning rate of 1e-5, max sequence length of 230, batch size of 16 and 3 epochs. Models with different task modules are optimized by selecting the checkpoint with the best Spearman correlation coefficient or Matthews Correlation Coefficient (MCC) on Dev set for each language pair separately. Three versions of PLM are pretrained as described in 3.4 for each combination of language pair and PLM, which are listed in the order of "DA-only, DA+MQM, HTER" in Table 4, Table 7 and Table 8 while Table 3 are only "DA-only, DA+MQM" for each language pair. Optuna (Akiba et al., 2019) is used to search the weights of model ensembles described in 3.5.

4.2 Results and Analysis

Sentence level For results in Table 3 of Dev set with MQM annotations, results based on mDeBERTa perform best in all settings. Models with PLMs pretrained on "DA-only" data achieve better results than those "DA+MQM" models which indicates that the difference in score range between DA and MQM has a great effect. For Table 7 of Dev set with DA annotations, models pretrained on "DA-only" data perform best among different combinations of PLMs and language pairs. Also, InfoXLM and XLM-Roberta-Large show higher correlations than other PLMs. Meanwhile, the score refinement defined in 3.2 has a positive impact in both Table 3 and Table 7 which suggests the necessity to unify the range of different scores. However, correlations of different PLMs vary a lot for each language pair which suggests we still have room for improvement. Also, when using multitask training, the Spearman correlation coefficient increases compared to only training on sentence-level data. The "DA+MQM" data improves the performance of En-De while becoming worse on Zh-En.

Word level The results in Table 4 indicate that pretraining data, PLM and task modules affect the model performance to varying degrees. Since HTER data is most related to word-level task, the

[6]https://github.com/huggingface/transformers

sentence level	MQM			
	En-De		Zh-En	
	DA-only			
XLM-Roberta-Large	0.5162	0.4219	0.3424	0.3028
mDeBERTa	0.5467	0.5281	0.3310	0.3717
RemBert	0.5040	0.4231	0.3048	0.2948
InfoXLM	0.5295	0.3786	0.3670	0.2881
	DA+MQM			
XLM-Roberta-Large	0.5346	0.4459	0.2889	0.2495
mDeBERTa	0.5668	0.5470	0.3110	0.3597
RemBert	0.5141	0.4323	0.3042	0.2822
InfoXLM	0.5342	0.4451	0.3039	0.2793
	DA-only w/ score_refine			
XLM-Roberta-Large	0.5218	0.4277	0.3254	0.2772
mDeBERTa	0.5435	0.5202	0.3319	0.3594
RemBert	0.5144	0.4092	0.2894	0.3080
InfoXLM	0.5266	0.4047	0.3734	0.2945
	DA+MQM w/ score_refine			
XLM-Roberta-Large	0.5386	0.4561	0.3005	0.2473
mDeBERTa	0.5728	0.5494	0.3227	0.3547
RemBert	0.5092	0.4302	0.2973	0.2840
InfoXLM	0.5309	0.4599	0.3037	0.2863

Table 3: Spearman correlation on Dev of sentence level on combinations of training data and score refinement(optional)

results based on pretraining on HTER data are best. Besides, models with RemBert or InfoXLM on En-De give bad results while models with BiLSTM as task module on Zh-En overfit on Dev set when submitting to test. In addition, swapping the order of *source* and *translation* has no improvement. For En-De and En-Mr, training on word-level data only is better than multitask training.

Multitask As shown in Table 8, multitask training improves the correlation of sentence-level task on all language pairs while only MCC of Zh-En grows. The score refinement method raises the correlation of word-level task obviously compared to models without applying score refinement. Yet, it does not always have a positive effect on sentence-level task. The multitask training for Zh-En avoids overfitting on Dev set and using BiLSTM as task module surpasses using FFN. Different PLMs will perform better if combined with specific task modules, which needs further experiments.

Ensemble The official results of models ensemble on dev and test for sentence level and word level are shown in Table 5 and Table 6 respectively. The ensemble method outperforms single model performance by a large margin. Our models have competitive results on all language pairs.

5 Conclusion

This paper describes our work for WMT 2023 Quality Estimation Task 1 on both sentence level and word level. With the help of PLMs and extra data, we can train better representations of *source* text and its *translation* for quality estimation task. We also experiment with diverse combinations of PLMs, task modules, and pretraining datasets. We find that QE systems for certain language pairs need to adopt particular combinations to acquire improvement, which reveals that there are distinct characteristics between languages. Such features make it hard to build one model for all languages, especially those without labeled data. The multitask training approach shows obvious improvements and prevents models from overfitting. Besides, the score refinement trick does not always give us positive feedback which suggests the number range is not the only factor to train on DA and MQM data properly. As expected, the ensemble method makes the predictions have a higher correlation with the ground truth. For future work, we will explore more profitable pretraining techniques for quality estimation and efficient modules that work well for various language pairs.

Limitations

Although our method has shown competitive results on most language pairs, evaluation results on zero-shot language pairs suggest that the model is not so powerful in generalization and relies on manual adjustment to some extent like choosing the weights among different models in the ensemble. Such operations could affect the model performance when transferring to unseen language pairs. Furthermore, we only designed two kinds of modules to generate tags in word-level task with slight improvement over baselines. It will be a potential research area to design more efficient prediction modules that can predict more accurate tags and we leave it as future work.

Also, other training configurations like weight decay and layer-wise learning rate decay were not experimented with sufficiently. Due to the discrepancy between training loss and evaluation metric, the choice of loss was a critical factor in model performance which was unexplored. Lastly, the limited amount of data constrained the improvement of models and overfitting on Dev set still has a great effect on optimization. We hope these analyses can promote the research of quality estimation.

word level	En-De			Zh-En			En-Mr		
	BiLSTM + regression(Eq. 13)								
mDeBERTa	0.3354	0.3364	0.3388	0.4483	0.4868	0.4447	0.3443	0.3500	0.3566
RemBert	0.0370	0.0160	0.0076	0.4842	0.4140	0.4736	0.3657	0.3601	0.3637
InfoXLM	0.0327	0.0456	0.0288	0.5491	0.4656	0.5249	0.3466	0.3385	0.3603
	FFN + regression(Eq. 14)								
mDeBERTa	0.3206	0.3306	0.3315	0.4666	0.5013	0.4727	0.3399	0.3396	0.3443
RemBert	0.3213	0.2477	0.3313	0.4715	0.4993	0.4575	0.3724	0.3158	0.3504
InfoXLM	0.2972	0.2905	0.3042	0.5411	0.4860	0.5230	0.3554	0.3407	0.3601
	FFN + regression(Eq. 14) w/ swap_order								
mDeBERTa	0.3167	0.3451	0.3306	0.5252	0.4951	0.4506	0.3305	0.3339	0.3469
RemBert	0.3123	0.2752	0.3023	0.4547	0.4513	0.4715	0.3610	0.3448	0.3153
InfoXLM	0.2969	0.2851	0.2957	0.5167	0.5032	0.5549	0.3520	0.3277	0.3626

Table 4: Spearman correlation on Dev of word level on combinations of tagging modules(BiLSTM/FFN) and regression modules with swapping orders(optional)

	Dev	Test
En-De	0.612	0.483
Zh-En	0.403	0.482
En-Mr	0.626	0.505
En-Gu	0.706	0.695
En-Hi	0.603	0.600
En-Ta	0.708	0.740
En-Te	0.474	0.376
He-En	-	0.575
Multilingual	-	0.513

Table 5: Spearman correlation of sentence level on Dev and Test

	Dev	Test
En-De	0.343	0.256
Zh-En	0.221	0.250
En-Mr	0.398	0.334
He-En	-	0.359
En-Fa	-	0.351
Multilingual	-	0.298

Table 6: MCC of word level on Dev and Test

Ethics Statement

This work follows all the rules of ACL Ethics Policy during the experiments of training and evaluation. The data used in this work are publicly available and widely used or provided by the organization of the competition. And to the best of the authors' knowledge, we do not foresee any risks against the ACL Ethics Policy.

Acknowledgements

The participants would like to express heartfelt thanks to the committee and the organizers of the WMT Quality Estimation Shared Task. We would also like to show our gratitude to the reviewers for their invaluable suggestions. This work is supported by Transn IOL Technology Co., Ltd.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Hyung Won Chung, Thibault Févry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Heidi Depraetere, Joachim Van den Bogaert, Sara Szoc, and Tom Vanallemeersch. 2020. APE-QUEST: an MT quality gate. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 473–474, Lisboa, Portugal. European Association for Machine Translation.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Lucia Specia and Atefeh Farzindar. 2010. Estimating machine translation post-editing effort with HTER. In *Proceedings of the Second Joint EM+/CNGL Workshop: Bringing MT to the User: Research on Integrating MT in the Translation Industry*, pages 33–43, Denver, Colorado, USA. Association for Machine Translation in the Americas.

Lucia Specia, Kim Harris, Frédéric Blain, Aljoscha Burchardt, Viviven Macketanz, Inguna Skadiņa, Matteo Negri, , and Marco Turchi. 2017. Translation quality and productivity: A study on rich morphology languages. In *Proceedings of Machine Translation Summit XVI*, pages 55–71, Nagoya, Japan.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 5998–6008. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4699–4709, Florence, Italy. Association for Computational Linguistics.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the WMT 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 69–99, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

sentence level	Direct Assessment														
	En-Mr			En-Gu			En-Hi			En-Ta			En-Te		
DA-only															
XLM-Roberta-Large	0.5867	0.5725	0.5814	0.6554	0.6367	0.6472	0.5152	0.5082	0.5207	0.6672	0.6658	0.6580	0.4092	0.4198	0.3972
mDeBERTa	0.5628	0.5467	0.5554	0.6370	0.6058	0.6288	0.5083	0.5004	0.5237	0.6423	0.6119	0.6538	0.3405	0.3263	0.3477
RemBert	0.5556	0.5593	0.5714	0.6649	0.6295	0.6422	0.5547	0.5350	0.5470	0.6598	0.6439	0.6592	0.4312	0.4028	0.3990
InfoXLM	0.5658	0.5697	0.5693	0.6715	0.6569	0.6693	0.5317	0.5386	0.5266	0.6701	0.6650	0.6708	0.4211	0.4131	0.3852
DA+MQM															
XLM-Roberta-Large	0.5869	0.5761	0.5582	0.6671	0.6352	0.6354	0.5282	0.5125	0.5268	0.6561	0.6694	0.6692	0.4202	0.4343	0.4293
mDeBERTa	0.5683	0.5591	0.5588	0.6275	0.5972	0.6252	0.5101	0.5043	0.5128	0.6307	0.6183	0.6244	0.3529	0.3251	0.3260
RemBert	0.5558	0.5802	0.5583	0.6357	0.6215	0.6541	0.5484	0.5401	0.5643	0.6507	0.6513	0.6657	0.4050	0.4094	0.4083
InfoXLM	0.5691	0.5713	0.5691	0.6631	0.6460	0.6609	0.5246	0.5227	0.5039	0.6792	0.6659	0.6685	0.4394	0.3978	0.4189
DA-only w/ score_refine															
XLM-Roberta-Large	0.5810	0.5740	0.5803	0.6738	0.6389	0.6614	0.5273	0.5307	0.5338	0.6730	0.6651	0.6764	0.4398	0.4174	0.3924
mDeBERTa	0.5599	0.5675	0.5539	0.6335	0.6187	0.6216	0.5022	0.5167	0.5245	0.6639	0.6318	0.6525	0.3476	0.3250	0.3449
RemBert	0.5888	0.5442	0.5688	0.6651	0.6378	0.6596	0.5826	0.5252	0.5813	0.6685	0.6554	0.6643	0.4166	0.3896	0.4348
InfoXLM	0.5705	0.5571	0.5814	0.6806	0.6678	0.6768	0.5485	0.5410	0.5417	0.6847	0.6704	0.6814	0.4256	0.4150	0.4086
DA+MQM w/ score_refine															
XLM-Roberta-Large	0.5740	0.5555	0.5773	0.6568	0.6308	0.6466	0.5205	0.4811	0.5065	0.6726	0.6506	0.6566	0.4256	0.4281	0.4176
mDeBERTa	0.5523	0.5756	0.5630	0.6434	0.6266	0.6236	0.5138	0.5170	0.5044	0.6412	0.6264	0.6330	0.3343	0.3447	0.3562
RemBert	0.5744	0.5620	0.5724	0.6387	0.6339	0.6328	0.5734	0.4939	0.5656	0.6433	0.6648	0.6618	0.4156	0.4161	0.4076
InfoXLM	0.5746	0.5724	0.5804	0.6746	0.6522	0.6627	0.5308	0.5367	0.5144	0.6894	0.6655	0.6634	0.4299	0.422	0.4488

Table 7: Spearman correlation on Dev of sentence level on combinations of training data and score refinement(optional)

multitask	En-De			Zh-En			En-Mr		
	SRM + BiLSTM								
mDeBERTa	0.5724/0.2933	0.5360/0.3503	0.5749/0.3012	0.3063/0.2360	0.3383/0.2062	0.3170/0.2412	0.5783/0.2294	0.5808/0.2381	0.5707/0.2086
RemBert	0.5390/0.0018	0.4353/0.0013	0.5340/0.0052	0.2882/0.1920	0.2661/0.1804	0.3108/0.1879	0.5895/0.2934	0.5710/0.2515	0.5926/0.3205
InfoXLM	0.5206/-0.0004	0.4342/0.0377	0.5033/0.0087	0.3237/0.2826	0.2791/0.2230	0.2980/0.2791	0.5712/0.2860	0.5739/0.2788	0.5801/0.3151
	SRM + FFN								
mDeBERTa	0.5638/0.3036	0.5389/0.3070	0.5738/0.3055	0.3113/0.2212	0.3377/0.2091	0.3140/0.2187	0.5859/0.2581	0.5669/0.2555	0.5829/0.2942
RemBert	0.5439/0.2475	0.4288/0.2509	0.5134/0.2869	0.2834/0.1774	0.2878/0.1558	0.3069/0.1516	0.5787/0.2982	0.5674/0.2615	0.5911/0.3036
InfoXLM	0.5317/0.3161	0.4254/0.1994	0.5358/0.2149	0.3487/0.3283	0.2721/0.2043	0.3114/0.2935	0.5703/0.3018	0.5668/0.2896	0.5799/0.3169
	SRM + BiLSTM w/ score_refine								
mDeBERTa	0.5580/0.3028	0.5411/0.3461	0.5597/0.2766	0.3281/0.2308	0.3438/0.2116	0.3143/0.2027	0.5877/0.2796	0.5709/0.2959	0.5834/0.2819
RemBert	0.5457/0.0080	0.4425/0.0119	0.5239/0.0051	0.2750/0.1871	0.2968/0.1695	0.3332/0.1684	0.5762/0.3176	0.5962/0.3233	0.5868/0.3250
InfoXLM	0.5272/0.0088	0.4186/0.0195	0.5097/-0.0042	0.3390/0.2763	0.2621/0.1742	0.3332/0.2704	0.5714/0.3062	0.5700/0.3026	0.5734/0.3227
	SRM + FFN w/ score_refine								
mDeBERTa	0.5681/0.3224	0.5364/0.3209	0.5706/0.3190	0.3277/0.2490	0.3364/0.1634	0.3052/0.2231	0.5841/0.2606	0.5735/0.2874	0.5803/0.3015
RemBert	0.5419/0.2808	0.4226/0.2376	0.5268/0.2688	0.2839/0.1695	0.2818/0.1663	0.3346/0.1743	0.5833/0.3086	0.5901/0.3194	0.5848/0.3254
InfoXLM	0.5062/0.2516	0.4300/0.2761	0.5072/0.2882	0.3256/0.2746	0.2677/0.2480	0.3059/0.2784	0.5679/0.3008	0.5753/0.3009	0.5809/0.3108

Table 8: Spearman correlation and MCC on Dev of multitask training on sentence and word levels of sentence regression module(SRM) with tagging modules(BiLSTM/FFN) and score refinement(optional)