Towards Making the Most of Pre-trained Translation Model for Quality Estimation

Chunyou Li[1], Hui Di[2], Hui Huang[1], Kazushige Ouchi[2], Yufeng Chen[1], Jian Liu[1], and Jinan Xu[1(✉)]

[1] Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing 100044, China
{21120368,chenyf,jianliu,jaxu}@bjtu.edu.cn
[2] Toshiba (China) Co., Ltd., Beijing, China
dihui@toshiba.com.cn, kazushige.ouchi@toshiba.co.jp

Abstract. Machine translation quality estimation (QE) aims to evaluate the quality of machine translation automatically without relying on any reference. One common practice is applying the translation model as a feature extractor. However, there exist several discrepancies between the translation model and the QE model. The translation model is trained in an autoregressive manner, while the QE model is performed in a non-autoregressive manner. Besides, the translation model only learns to model human-crafted parallel data, while the QE model needs to model machine-translated noisy data. In order to bridge these discrepancies, we propose two strategies to post-train the translation model, namely Conditional Masked Language Modeling (CMLM) and Denoising Restoration (DR). Specifically, CMLM learns to predict masked tokens at the target side conditioned on the source sentence. DR firstly introduces noise to the target side of parallel data, and the model is trained to detect and recover the introduced noise. Both strategies can adapt the pre-trained translation model to the QE-style prediction task. Experimental results show that our model achieves impressive results, significantly outperforming the baseline model, verifying the effectiveness of our proposed methods.

Keywords: Quality estimation · Machine translation · Denoising restoration

1 Introduction

Machine translation has always been the hotspot and focus of research. Compared with traditional methods, neural machine translation (NMT) has achieved great success. However, current translation systems are still not perfect to meet the real-world applications without human post-editing. Therefore, to carry out risk assessment and quality control for machine translation, how to evaluate the quality of machine translation is also an important problem.

M. Sun et al. (Eds.): CCL 2022, LNAI 13603, pp. 197–211, 2022.
https://doi.org/10.1007/978-3-031-18315-7_13

Quality Estimation (QE) aims to predict the quality of machine translation automatically without relying on reference. Compared with commonly used machine translation metrics such as BLEU [18] and METEOR [13], QE can be applicable to the case where reference translations are unavailable. It has a wide range of applications in post-editing and quality control for machine translation. The biggest challenge for QE is data scarcity. Since QE data is often limited in size, it is natural to transfer bilingual knowledge from parallel data to the QE task.

One well-known framework for this knowledge transfer is the predictor-estimator framework, in which the predictor is trained on large parallel data and used to extract features, and the estimator will make quality estimation based on features provided by the predictor. The predictor is usually a machine translation model, which can hopefully capture the alignment or semantic information of the source and the target in a pair. Kim et al. [11] first proposed to use an RNN-based machine translation model as the feature extractor, to leverage massive parallel data to alleviate the sparsity of annotated QE data. Wang et al. [23] employed a pre-trained translation model as the predictor and added pseudo-PE information to predict translation quality.

However, there are two discrepancies between machine translation and quality prediction, which impedes the NMT model to be directly adopted for feature extraction. i) Translation task is usually a language generation task trained in an autoregressive manner, where each token is only conditioned on previous tokens unidirectionally. But QE is a language understanding task performed in a non-autoregressive manner, therefore each token could attend to the whole context bidirectionally. ii) The predictor is trained on human-crafted parallel data and only learns to model the alignment between correct translation pairs. However, the QE task needs to model machine-translated, imperfect translation pairs. Both discrepancies may hinder the adaptation of the pre-trained NMT model to the downstream QE task, leading a degradation of model performance [25].

In this paper, we propose two strategies to alleviate the discrepancies, named as Conditional Mask Language Modeling (CMLM) and Denoising Restoration (DR). Both strategies are applied to the pre-trained NMT model and can be deemed as a post-training phase. The CMLM is to train the NMT model to recover the masked tokens at the target side in a non-autoregressive manner, where each token can attend to the whole target sequence bidirectionally. Furthermore, the DR first generates erroneous translation by performing conditionally masked language modeling, and then trains the NMT model to detect the introduced noise and recover the target sequence, which is also performed in a non-autoregressive manner. Both methods can adapt the autoregressive NMT model to non-autoregressive QE prediction. Moreover, compared with CMLM, DR removes the introduction of [MASK] token (which may also cause the discrepancy between pre-training and QE prediction). Besides, adversarially using another model with knowledge distillation to generate noise could provide more natural and harder training samples, thereby pushing the translation model better model the semantic alignment between the imperfect translation and source

sequence. After the post-training phase, the NMT model is better adapted to the quality prediction task, and can serve as a better feature extractor.

Our contributions can be summarized as follows:

- We propose two strategies for post-training the NMT model to bridge the gaps between machine translation and quality estimation, which can make the NMT model more suitable to act as the feature extractor for the QE task.
- We conduct experiments on the WMT21 QE tasks for En-Zh and En-De directions, and our methods outperform the baseline model by a large margin, proving its effectiveness. We also perform in-depth analysis to dig into the discrepancies between translation and quality prediction.

2 Background

2.1 Task Description

Quality Estimation aims to predict the translation quality of an MT system without relying on any reference. In this task, the dataset is expressed in the format of triplet (s, m, q), where s represents the source sentence, m is the translation output from a machine translation system, and q is the quality score of machine translation.

Generally, Quality Estimation task includes both word-level and sentence-level tasks. In word-level task, the prediction is done both on source side (to detect which words caused errors) and target side (to detect mistranslated or missing words). In sentence-level task, it will mark each sentence with a score, which can be calculated based on different standards, consists of Human-targeted Translation Edit Rate (HTER) [21], Direct Assessment (DA) [8], Multidimensional Quality Metrics (MQM) [15], etc. In this work, we mainly focus on sentence level post-editing effort prediction, which is measured by:

$$HTER = (I + D + R)/L, \tag{1}$$

where I, D and R are the number of Insertions, Deletions and Replacement operations required for post-editing, and L is the reference length. However, labeling the data requires post-editing for the machine translations by experts, leading the label of QE data too expensive to obtain, which makes QE highly data-sparse.

2.2 Previous Work

Generally, sentence-level QE is fomulated as a regression task. Early approaches were based on features fed into a traditional machine learning method, such as QuEst++ [22] and MARMOT [14] system. These model usually has two modules: the feature extraction module and the classification module. But they relied

on heuristic artificial feature designing, which limits their development and application [10]. With the increasing popularity of deep learning methods, researchers resort to distributed representations and recurrent networks to encode translation pairs. However, the limited size of training samples impedes the learning of deep networks [16]. To solve this problem, a lot of research has been done to use additional resource (both bilingual and monolingual) to strengthen the representation [11]. After the emergence of BERT [5], some work attempts to use the pre-trained language model as a predictor directly and add a simple linear on top of the model to obtain the predictions [1,2], which has led to significant improvements.

Among all the deep learning-based methods, one commonly used framework for QE is the predictor-estimator framework, where the predictor is used as a feature extractor and the estimator uses the features to make predictions. The predictor is usually a translation model, which can alleviate the problem of data sparsity by transferring bilingual knowledge from parallel data. Kim et al. [11] firstly proposed the predictor-estimator framework to leverage massive parallel data to improve QE results, they applied an RNN-based machine translation model as the predictor and added a bidirectional RNN as estimator to predict QE scores, which achieved excellent performance especially in sentence-level QE. Fan et al. [6] used Transformer-based NMT model as the predictor to extract high-quality features, and used 4-dimensional mis-matching features from this model to improve performance. Wang et al. [24] pre-trained left-to-right and right-to-left deep Transformer models as the predictor and introduced a multi-layer bidirectional Gated Recurrent Unit (Bi-GRU) as the estimator to make prediction. Wu et al. [26] reformed Transformer-based predictor-estimator by using multidecoding during the machine translation module, then implemented LSTM-based and Transformer-based estimator with top-K and multi-head attention strategy to enhance the sentence feature representation. Wang et al. [23] employed a pre-trained translation model as the predictor and added pseudo-PE information to predict translation quality, which obtained the best result in the English-German direction of WMT20. However, despite various of improvement has been made on the predictor-estimator framework, the discrepancy problem between machine translation and quality estimation is not systematically investigated.

3 Approach

In this section, we first describe the NMT-based QE architecture, and then describe our proposed post-training strategies.

3.1 QE Architecture

The QE architecture is shown in Fig. 1. Our work follows the predictor-estimator framework. The predictor is a translation model trained with the transformer architecture on parallel data, which has learned the feature extraction ability of

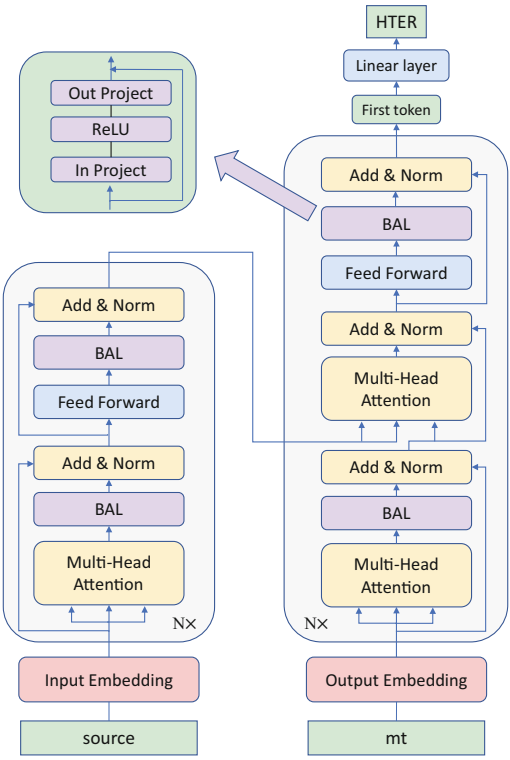

Fig. 1. The illustration of the QE model. The *source* and *mt* sentence are fed into encoder and decoder respectively. The BAL is integrated after the self-attention layer and the FFN layer, respectively. In order to better adapt to QE task, the causal mask in decoder is removed.

bilingual inputs after a long-term and large-scale pre-training. Therefore, adding only a linear layer on the top of translation model and fine-tuning with a small amount of QE data can achieve promising results.

As shown in Fig. 1, the final hidden vector of the neural machine translation model corresponding to the first input token is fed into a simple linear layer to make quality prediction, which is given by:

$$HTER_{pred} = W_s^T h^{(0)} + b_0, \tag{2}$$

where $h^{(0)} \in \mathbb{R}^H$ is the hidden vector of the first input token, $W_s \in \mathbb{R}^H$ represents a weight matrix, H is the dimension of hidden states, $b_0 \in \mathbb{R}^1$ is the bias. The loss function is the mean squared error between $HTER_{pred}$ and $HTER_{true}$, which can be written as:

$$L_{QE} = MSE(HTER_{pred}, HTER_{true}) \tag{3}$$

Since the size of training dataset is relatively small, the model is easy to be over-fitted when all parameters are updated. Incorporating the insights from Wang et al. [23], the Bottleneck Adapter Layers (BAL) [9] are integrated into the neural machine translation model, which alleviates the problem of overfitting by freezing the parameters of the original model. The BAL is implemented with two simple fully-connected layers, a non-linear activation and residual connections, where the hidden representations are first expanded two times and then reduced back to the original dimension.

3.2 Conditional Masked Language Modeling

The Conditional Masked Language Modeling is illustrated in Fig. 2. Despite using the same architecture as the machine translation model, the CMLM utilizes a mask language modeling objective at the target side [7]. The source sentence is sent to the encoder, while some tokens are corrupted at the target side. Then the CMLM is trained to recover the corrupted target sentence.

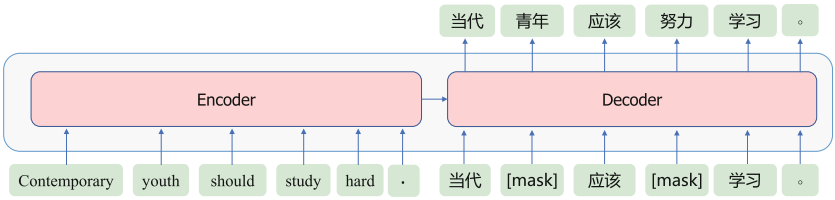

Fig. 2. The illustration of the CMLM. At the target side, some tokens are replaced with [mask] symbol or random token. Note that it also needs to remove the casual mask in decoder.

In terms of implementation, given a parallel sentence pair $<x, y>$, we generate a corrupted sentence y' with a 25% mask ratio. When the i-th token is chosen to be masked, it may be replaced with the [MASK] token 20% of the time or a random token 80% of the time. The training objective for CMLM is to maximize: $P(y_i|x, y')$, where y_i is the i-th token, x and y' represent the source sentence and the corrupted target sentence, respectively. More specifically, we reuse the parameters of the neural machine translation model instead of training the model from scratch, and the model is trained with data in the same domain as the QE data.

Translation model is a natural language generation model trained in an autoregressive manner, where each token can only pay attention to the tokens before it, and the tokens after it are masked out. On the contrary, QE task is a natural language understanding task in which each token needs to be concerned with the whole context. Through this mask-prediction task focusing on bidirectional information, the model can learn the context-based representation of the

token at the target side, thereby adapting the unidirectional NMT decoder to the bidirectional prediction task.

3.3 Denoising Restoration

Inspired by Electra [3], to further mitigate the discrepancy of data quality, we apply the Denoising Restoration strategy to post-train the neural machine translation model. The model architecture is illustrated in Fig. 3, which can be divided into the Noiser and the Restorer. The Noiser is used to create noisy samples, and the restorer is used to recover the noisy samples. After that, only the Restorer would be used as the predictor and the Noiser would be dropped.

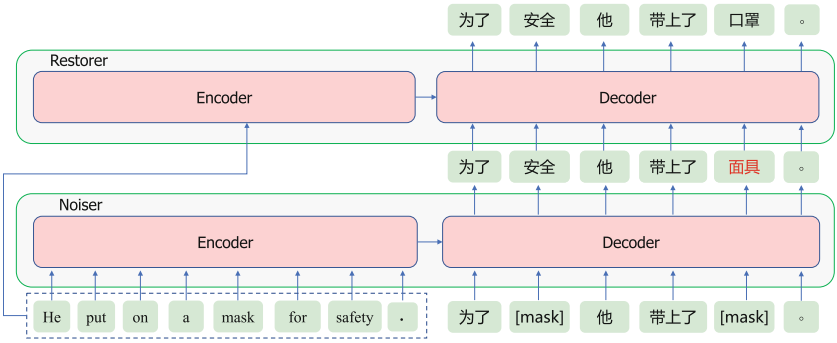

Fig. 3. The Noiser-Restorer architecture.

The Noiser is first trained to introduce noise at the target side. It has the same architecture as the CMLM, the difference is that we utilize the decoding results of the to-be-evaluated NMT model as the training objective of the Noiser, where the to-be-evaluate NMT model is used to generate QE data. Specifically, given a parallel sentence pair $<x, y>$, we use the to-be-evaluted NMT model to generate the translation $\tilde{y}$ of x. Then the Noiser is trained with the new parallel sentence pair $<x, \tilde{y}>$. After the training of the Noiser, we put the Noiser and the Restorer together for training with parallel data$<x, y>$. Moreover, it is performed by dynamic mask strategy with the masked positions decided on-the-fly, where the mask ratio is same as that of the CMLM. The loss function is defined as follows:

$$L_{DR} = -\sum_{i=1}^{L} logP(l = l_i|x, \hat{y}), l_i \in \{1, 2, ..., V\}, \tag{4}$$

where L is the length of sentence, $\hat{y}$ is the sentence generated by the Noiser, V is the size of vocabulary.

The reason for introducing Noiser is that in the CMLM strategy, there is a large deviation between the sentences generated by randomly adding noise and

real machine translation, which is easily detected and may limit the performance. Limited by the performance of the Noiser, it is certain that not all tokens can be recovered completely and correctly. Therefore, the target sequence generated by the Noiser is noisy compared with reference translation. Meanwhile, since the Noiser utilizes a decoder with language modeling capabilities for generation, the generated sentences are more natural without obvious lexical and syntactic errors. Similarly, real machine translation noise is also natural and does not have significant lexical and syntactic errors, so the noise generated by the Noiser is closer to the real noise distribution than the noise generated by random replacement. A possible example is shown in the Fig. 3.

In addition, we utilize knowledge distillation technique [12] in the Noiser, which is used to transfer specific patterns and knowledge among different sequence generation models. In our scenario, the decoding process of the to-be-evaluated NMT model has a fixed pattern, so the translation results obtained by decoding the source sentences with this NMT model contains the noise distribution of the to-be-evaluated NMT model. When the Noiser learns to recover a corrupted token, both training objectives and context are generated by this NMT model. Hence, the obtained Noiser would have a similar decoding space with the to-be-evaluated NMT model. Note that the Noiser could produce pseudo translations with the same length as the reference translation, which is convenient for later training.

Despite both adopting non-autoregressive training objective, the difference between CMLM and Restorer lies in the source of noise. The noise of CMLM comes from random masking, while the noise of Restorer comes from language model generation. On the one hand, the noise generated by the Noiser is more consistent with the noise distribution of the to-be-evaluated NMT model, so during the training, the Restorer can learn the modeling ability for noise data with specific distribution. On the other hand, since the noise generated by the Noiser is more natural and more difficult to identify, the obtained Restorer would have a better feature extraction ability and can identify trivial translation errors. In cases where QE needs to model machine-translated noisy data, the Restorer is more suitable for QE task.

4 Experiments

4.1 Settings

Dataset. Our experiments focus on the WMT21 QE tasks for English-to-Chinese (En-Zh) and English -to-German (En-De) directions. The QE data in each direction contains a training set of 7000, a validation set of 1000, and a test set of 1000. Besides, we also use the test set of WMT20. To train our own NMT model, we use the En-Zh and En-De parallel data released by the organizers[1], which contains roughly 20M sentence pairs for each direction after cleaning. For the CMLM and DR, We first trained a BERT-based domain classifier and

[1] https://www.statmt.org/wmt21/quality-estimation-task.html.

then screened 200K in-domain data from WikiMatrix for each direction[2]. The validation set we use is the training set of the QE task.

Implementation Details. All our programs are implemented with Fairseq [17]. For the NMT model, we use Transformer-base architecture. We apply byte-pair-encoding (BPE) [20] tokenization to reduce the number of unknown tokens and set BPE steps to 32000. The learning rate is set to 5e−4. This setting is adopted in both En-Zh and En-De directions.

For the CMLM, the casual mask is removed and learning rate is set to 5e-5. For the Noiser-Restorer model, the parameters of the Noiser are frozen and the learning rate for the Restorer is 5e−5. For the Noiser, we use the decoding results of the to-be-evaluated NMT model as the training objective. We use inverse-square-root scheduler in above three models. For the QE model, it trained for 30 epochs and the hyperparameter patience is set to 5. The activation function in the BAL is ReLU. We batch sentence pairs with 4096 tokens and use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-8}$. The learning rate is 1e-4 without any scheduler.

The training data for all models is preprocessed by Fairseq based on the vocabulary and BPE vocabulary of the NMT model. For fair comparison, we tune all the hyper-parameters of our model on the validation data, and report the corresponding results for the testing set. The main metric we use is Pearson's Correlation Coefficient. We also calculate Spearman Coefficient, but it is not a ranking reference in the QE task.

4.2 Main Results

We compare our models with the following methods:

PLM-Baseline: Pre-training language models (PLM) are directly used as the predictor without integrating the BAL layer. In our experiments, DistilBert [19] and XLM-RoBERTa [4] were selected, and the baseline of organisers is also implemented by XLM-RoBERTa.

NMT-Baseline: An NMT model pre-trained on parallel data is used as the predictor, where NMT(finetune) is obtained by continuing to finetune on the in-domain data used by CMLM and DR.

The experimental results in both En-Zh and En-De directions are reported in Table 1. The Test20 is officially corrected, so there are no up-to-date results. As can be seen, the performance of the baseline model is relatively poor. By leveraging MLM training strategies, the CMLM can better focus on contextual information and achieves much better performance than NMT model. Moreover, the denoising restoration strategy further enhances the feature extraction ability of Restorer by introducing noise that is consistent with the distribution of NMT and outperforms the CMLM in two language pairs. This illustrates that our approaches alleviate the discrepancy between the NMT model and the QE model,

[2] http://data.statmt.org/wmt21/translation-task/WikiMatrix.

Table 1. Experiment results on both En-Zh and En-De directions. 'XLM-R' and 'DistilBERT' are implemented by us based on XLM-RoBERTa and DistilBERT. 'Avg' represents the average value of the pearson over two datasets. '-' indicates missing results.

Direction	System	Test21		Test20		Avg
		Pearson↑	Spearman↑	Pearson↑	Spearman↑	
En-Zh	XLM-R (WMT-baseline)	0.282	–	–	–	0.282
	DistilBert	0.257	0.223	0.340	0.334	0.299
	XLM-R	0.265	0.219	0.323	0.318	0.294
	NMT	0.286	0.242	0.322	0.312	0.304
	NMT (finetune)	0.294	0.243	0.322	0.311	0.308
	CMLM	0.334	0.273	0.355	0.345	0.345
	DR	**0.342**	**0.275**	**0.362**	**0.353**	**0.352**
En-De	XLM-R (WMT-baseline)	0.529	–	–	–	0.529
	DistilBert	0.466	0.433	0.432	0.427	0.449
	XLM-R	0.537	0.492	**0.469**	**0.464**	0.503
	NMT	0.528	0.491	0.427	0.424	0.478
	NMT (finetune)	0.532	0.491	0.438	0.430	0.485
	CMLM	0.569	0.518	0.450	0.437	0.509
	DR	**0.577**	**0.521**	0.460	0.424	**0.519**

thereby making the NMT model better adapted to the QE task. Combined with the official ranking, in En-Zh direction, our single model outperforms other systems except the first place (which adapt multiple ensemble techniques and data-augmentation).

The CMLM and DR also perform better than the fine-tuned NMT model, which indicates the performance gains of them are not due to the introduction of additional datasets. Besides, the NMT-based models are more effective than PLM-Baseline in most of the comparisons, we consider that the NMT model is naturally fit for machine translation related tasks, benefiting from the knowledge of bilingual alignment.

5 Analysis

5.1 The Impact of Mask Ratio and [MASK] Symbol

During the training stage, the number of corrupted tokens may affect the performance of the model, which is related to the mask ratio. We conduct experiments to study the impact of different mask ratio and the results are illustrated in Fig. 4.

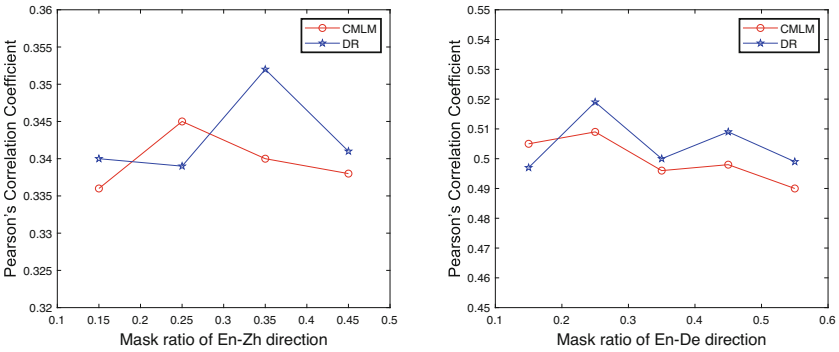

Fig. 4. The illustration of the CMLM. At the target side, some tokens are replaced with [mask] symbol or random token. Note that it also needs to remove the casual mask in decoder.

We find that the two diagrams exhibit roughly the same pattern. The QE performance first improves, but when the mask ratio is too high, the results start to decline. This is because as the mask ratio increases, the quality of the pseudo data is gradually approaching the real machine translation, therefore the model can better model semantic alignment between the imperfect translation and source. However, when the mask ratio is too high, most of the input sentence is covered and it is too difficult for the model to restore them, thus the model can barely learn anything useful and the performance is degraded. We also observe that the performance peak of the Noiser-Restorer model in En-Zh direction comes later than that in the En-De direction. One possible reason is that the Noiser in the En-Zh direction performs better than that in the En-De direction, we will explain this in the next subsection.

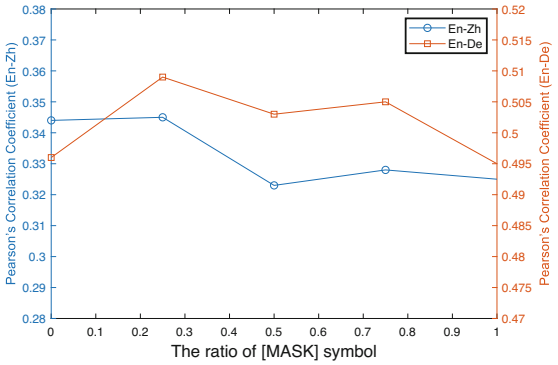

Fig. 5. The impact of the [MASK] symbol.

In the CMLM strategy, among the corrupted tokens, some will be replaced with [MASK] symbol, and the others will be replaced with random tokens. We fix the mask ratio and then gradually increase the proportion of corrupted tokens replaced with [MASK] symbol to study the impact of introducing [MASK] symbol. The results are presented in Fig. 5. We can observe that performance get worse as the introduced [MASK] symbol increases. It may be caused by the mismatch between pre-training and fine-tuning when too many [MASK] tokens are introduced, as they never appear during the fine-tuning stage. Furthermore, using only random replacement does not give the best results, which proves that the performance improvement brought by DR is not only due to the removal of [MASK] symbol but also benefits from the introduction of natural noise close to the real machine translation.

5.2 The Impact of Knowledge Distillation

In the implementation of the Noiser, we use the decoding results of the to-be-evaluated NMT model as the training objective of the Noiser. Our motivation is to make the Noiser learn the knowledge implied by to-be-evaluated model, so as to generate sentences that is closer to the noise of real machine translation. We conduct experiments to verify the effective of this scheme, and the results are shown in Table 2.

Table 2. The comparison results of Noiser-Restorer under two strategies. '*w/* kd' and '*w/o* kd' denote with or without knowledge distillation, respectively. The 'MAE' is the Mean Absolute Error.

Direction	System	Test21		Test20		Avg
		Pearson↑	MAE↓	Pearson↑	MAE↓	
En-Zh	Noiser-Restorer *w/o* kd	0.328	0.240	0.346	**0.226**	0.337
	Noiser-Restorer *w/* kd	**0.334**	**0.202**	**0.360**	0.233	**0.347**
En-De	Noiser-Restorer *w/o* kd	0.546	**0.125**	**0.449**	0.144	**0.498**
	Noiser-Restorer *w/* kd	**0.549**	0.128	0.436	**0.133**	0.493

For a fair comparison, we extracted another dataset from WikiMatrix instead of the one used to train the Noiser for experiments. According to the experimental results, we find that the scheme plays an obvious role in the En-Zh direction, which shows that the Noiser generates pseudo data consistent with the noise distribution of the to-be-evaluated NMT model, thereby improving the performance. However, the situation is different for the En-De direction, where the results are not improved or even slightly decreased as a whole. We speculate that it may be affected by the performance of the to-be-evaluated neural machine translation model. We studied the QE dataset and came up with the results shown in the Table 3.

Table 3. The statistical results of translation quality for QE dataset in En-Zh and En-De directions. The values in the table represent the average value of hter label.

Direction	train	valid	test21	test20
En-Zh	0.4412	0.2839	0.2283	0.3329
En-De	0.1784	0.1830	0.1754	0.1667

HTER indicates human-targeted translation edit rate, and the higher HTER is, the worse the translation quality is. As can be seen in Table 3, the average value of HTER in the En-Zh direction is generally higher than that in the En-De direction. This shows that the to-be-evaluated NMT model has a better translation effect in the En-De direction, thus the machine translation is not much different from the reference translation. It is difficult for Noiser to learn the pattern contained in the NMT model, so the knowledge distillation does not play a significant role.

5.3 Different Loss Calculation Methods

Base on previous researches, there are two ways to calculate the loss:

i. Following BERT, calculating the loss only on the small subset that was masked out.
ii. Calculating the loss over all input tokens at the target side.

Table 4. Experimental results of different loss calculation methods in En-Zh and En-De directions. 'Only-Corrupted' and 'All-Tokens' mean the loss is calculated on the corrupted tokens and all input tokens, respectively.

Direction	System	Test21		Test20		Avg
		Pearson↑	MAE↓	Pearson↑	MAE↓	
En-Zh	Only-Corrupted	0.328	0.217	0.348	**0.227**	0.338
	All-Tokens	**0.334**	**0.202**	**0.355**	0.233	**0.345**
En-De	Only-Corrupted	**0.574**	**0.125**	0.445	0.136	**0.510**
	All-Tokens	0.568	0.126	**0.450**	**0.132**	0.509

We compare these two methods on the CMLM strategy and the results are shown in Table 4. In the En-Zh direction, the method of calculating the loss on all tokens is better than that only on the corrupted tokens. However, the situation is a little different in the En-De direction. We speculate that English and German belong to the same family of languages, and the prediction is relatively simple, so adding this additional information has little effect. Overall, the performance of the two methods is roughly equivalent.

6 Conclusion

When applying the pre-trained machine translation model to feature extraction for QE, there are two discrepancies between the NMT model and the QE model. One is the difference in data quality, the other is the regressive behavior of the decoder. In this paper, we propose two strategies to adapt the neural machine translation model to QE task, namely Conditional Masked Language Modeling and Denoising Restoration. The CMLM adopts a mask-prediction task at the target side, which allows the model to learn context-based representations. Moreover, the DR employs a Noiser-Restorer architecture, where the Noiser is used to generate sentences with the same noise distribution as machine translation, then the Restorer will detect and recover the introduced noise. Compared with the original NMT model, our methods bridge the gaps between the NMT model and the QE model, making it more suitable for the QE task. The experimental results verify the effectiveness of our methods.

The main work in this paper focuses on sentence-level task. Intuitively, the discrepancy also exists on word-level quality estimation when applying the pre-trained NMT model, and our strategies could function without any adaptation. Besides, enhancing the estimator can also improve QE performance, and we will leave this as our future work.

Acknowledgements. The research work descried in this paper has been supported by the National Key R&D Program of China (2020AAA0108001) and the National Nature Science Foundation of China (No. 61976015, 61976016, 61876198 and 61370130). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

1. Chen, Y., et al.: HW-TSC's participation at WMT 2021 quality estimation shared task. In: Proceedings of the Sixth Conference on Machine Translation, pp. 890–896 (2021)
2. Chowdhury, S., Baili, N., Vannah, B.: Ensemble fine-tuned mBERT for translation quality estimation. arXiv preprint arXiv:2109.03914 (2021)
3. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)
4. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116 (2019)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Fan, K., Wang, J., Li, B., Zhou, F., Chen, B., Si, L.: "Bilingual expert" can find translation errors. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6367–6374 (2019)
7. Ghazvininejad, M., Levy, O., Liu, Y., Zettlemoyer, L.: Mask-predict: parallel decoding of conditional masked language models. arXiv preprint arXiv:1904.09324 (2019)

8. Graham, Y., Baldwin, T., Moffat, A., Zobel, J.: Can machine translation systems be evaluated by the crowd alone. Nat. Lang. Eng. **23**(1), 3–30 (2017)
9. Houlsby, N., et al.: Parameter-efficient transfer learning for NLP. In: International Conference on Machine Learning, pp. 2790–2799. PMLR (2019)
10. Huang, H., Di, H., Xu, J., Ouchi, K., Chen, Y.: Unsupervised machine translation quality estimation in black-box setting. In: Li, J., Way, A. (eds.) CCMT 2020. CCIS, vol. 1328, pp. 24–36. Springer, Singapore (2020). https://doi.org/10.1007/978-981-33-6162-1_3
11. Kim, H., Lee, J.H.: Recurrent neural network based translation quality estimation. In: Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, pp. 787–792 (2016)
12. Kim, Y., Rush, A.M.: Sequence-level knowledge distillation. arXiv preprint arXiv:1606.07947 (2016)
13. Lavie, A., Denkowski, M.J.: The meteor metric for automatic evaluation of machine translation. Mach. Transl. **23**(2), 105–115 (2009)
14. Logacheva, V., Hokamp, C., Specia, L.: Marmot: a toolkit for translation quality estimation at the word level. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), pp. 3671–3674 (2016)
15. Lommel, A., Uszkoreit, H., Burchardt, A.: Multidimensional quality metrics (MQM): a framework for declaring and describing translation quality metrics. Rev. Tradumàtica: Tecnol. Traducció **12**, 455–463 (2014)
16. Martins, A.F., Junczys-Dowmunt, M., Kepler, F.N., Astudillo, R., Hokamp, C., Grundkiewicz, R.: Pushing the limits of translation quality estimation. Trans. Assoc. Comput. Linguist. **5**, 205–218 (2017)
17. Ott, M., et al.: fairseq: a fast, extensible toolkit for sequence modeling. arXiv preprint arXiv:1904.01038 (2019)
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
19. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
20. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909 (2015)
21. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, pp. 223–231 (2006)
22. Specia, L., Paetzold, G., Scarton, C.: Multi-level translation quality prediction with quest++. In: Proceedings of ACL-IJCNLP 2015 system demonstrations, pp. 115–120 (2015)
23. Wang, M., et al.: HW-TSC's participation at WMT 2020 quality estimation shared task. In: Proceedings of the Fifth Conference on Machine Translation, pp. 1056–1061 (2020)
24. Wang, Z., et al.: Niutrans submission for CCMT19 quality estimation task. In: China Conference on Machine Translation, pp. 82–92. Springer (2019)
25. Weiss, K., Khoshgoftaar, T.M., Wang, D.D.: A survey of transfer learning. J. Big Data **3**(1), 1–40 (2016). https://doi.org/10.1186/s40537-016-0043-6
26. Wu, H., et al.: Tencent submission for WMT20 quality estimation shared task. In: Proceedings of the Fifth Conference on Machine Translation, pp. 1062–1067 (2020)