DAQE: Exploring the Direct Assessment on Word-Level Quality Estimation in Machine Translation

Anonymous ACL submission

Abstract

Word-level Quality Estimation (QE) of Machine Translation (MT) helps to find out potential translation errors in translated sentences without reference. The current collection of QE datasets is typically based on the exact matching between the words from MT sentences and post-edited sentences through a Translation Error Rate (TER) toolkit. However, we find that the data generated by TER cannot faithfully reflect human judgment, which may make the research deviate from the correct direction. To overcome the limitation, we for the first time collect the direct assessment (DA) dataset for the word-level QE task, namely DAQE, which is a golden corpus annotated by expert translators on two language pairs. Furthermore, we propose two tag correcting strategies, namely tag refinement strategy and tree-based annotation strategy, to make the TER-based artificial QE tags closer to human judgement, so that the automatically corrected and large-scale TER-based data can be used to improve the QE performance by pre-training. We conduct detailed experiments on our collected DAQE dataset, as well as comparison with the TER-based QE dataset MLQE-PE. The results not only show our proposed dataset DAQE is more consistent with human judgment but also confirm the effectiveness of the tag correcting strategies.[1]

1 Introduction

Quality Estimation (QE) of Machine Translation (MT) aims to automatically estimate the quality of the translation generated by MT systems, with no reference available. It typically acts as a post-processing module in commercial MT systems, determining whether the translation needs to be post-edited or alerting the user with potential translation errors. Recently, with the success of neural networks, neural-based QE models have achieved remarkable performance (Kepler et al., 2019; Kim et al., 2017; Lee, 2020; Specia et al., 2020; Ranasinghe et al., 2020; Wang et al., 2020b).

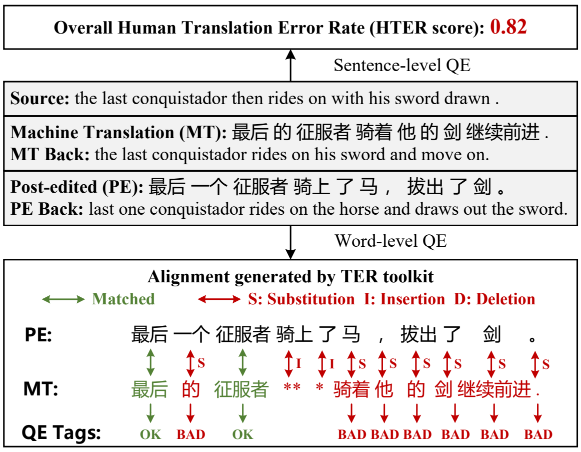


Figure 1: The illustration of the sentence-level and word-level QE tasks. The word-level QE tags are generated by the TER toolkit.

Figure 1 shows an example of QE. The sentence-level task predicts a score indicating the overall translation quality, while the word-level QE needs to annotate each word as OK or BAD[2]. Currently, the collection of QE datasets mainly relies on the Translation Error Rate (TER) toolkit (Snover et al., 2006). Specifically, given the machine translations and their corresponding post-edits (PE, generated by human translators) or target sentences of parallel corpus as the pseudo-PE (Tuan et al., 2021; Lee, 2020), the rule-based TER toolkit is used to generate the word-level alignment between the MT and the PE based on the principle of minimal editing. All MT words not aligned to PE are annotated as BAD (shown in Figure 1). Such annotation is also referred as post-editing effort (Fomicheva et al., 2020; Specia et al., 2020).

Although the TER-based annotation can auto-

[1] The codes and data samples are attached as supplementary materials. Our codes with the full data will be publicly available once accepted.

[2] In this paper, we mainly focus on the word-level QE on the target side, while we also show in our experiment that sentence-level QE can be implemented through the word-level QE.

Source: It is happy for me to be asked to speak here.

MT: 我 很 高兴 被 要求 在 这里 发言 。 **MT Back:** I am so happy to be asked to speak here.

PE: 被 邀请 在 这里 讲话 我 很 高兴 。 **PE Back:** Being invited to talk here makes me so happy.

TER-based Annotations: 我 很 高兴 被 要求 在 这里 发言 。
Human's Direct Assessment (DA): 我 很 高兴 被 要求 在 这里 发言 。

a) Some words in MT are mistakenly annotated to **BAD** though the overall semantic is not changed.

Source: The Zaporizhian Hetman was then dispatched to Istanbul, and impaled on hooks.

MT: 扎 波罗 齐安海 特曼 号 随后 被 派 往 伊斯坦布尔 ， 并 被 撞 在 钩 上 。
MT Back: The Zaporizhian Hetman was then dispatched to Istanbul, and was bumped on the hook.

PE: Zaporizhian Hetman 随后 被 派 往 伊斯坦布尔， 并 被 钉 在 钩子 上 。
PE Back: Zaporizhian Hetman was then dispatched to Istanbul, and was nailed on hooks.

TER-based Annotations: 扎 波罗 齐安海 特曼 号 随后 被 派 往 伊斯坦布尔 ， 并 被 撞 在 钩 上 。
Human's Direct Assessment (DA): 扎 波罗 齐安海 特曼 号 随后 被 派 往 伊斯坦布尔 ， 并 被 撞 在 钩 上 。

b) Human's DA annotates the clause "被撞在钩上" as a whole, while TER-based annotations are fragmented.

Figure 2: Two examples show the gap between the TER-based annotation and human's direct assessment on word-level QE task. The red color indicates BAD tags, while the green color indicates OK tags.

matically generate large-scale artificial QE data, we find two issues that make it inconsistent with human judgment. First, the PE sentences often substitute some words with better synonyms and reorder some sentence constituents for polish purposes. These operations do not destroy the translation semantics, but make some words mistakenly annotated under the exact matching criterion of TER. (shown in Figure 2a). Second, when fatal errors occur in MTs, a human's DA typically annotates the whole sentence or clause as BAD. However, TER-based annotations still try to find trivial words that align with PE, resulting in fragmented annotations (shown in Figure 2b). The WMT20 QE shared task includes the DA on the sentence-level QE as a subtask (Fomicheva et al., 2020), but it neglects the DA on the word-level QE. Meanwhile, most previous works still use the TER-based dataset as the evaluation benchmark of the word-level QE task. Their experimental results may not truly reflect the model's ability on finding translation errors, making the research deviate from the correct direction. Thus, there is an urgent need for a DA dataset that can precisely reflect human judgment on the word-level QE.

To overcome the limitations stated above, for the first time, we concentrate on the direct assessment of the word-level QE task. We first collect a new QE dataset called DAQE that reflects human's direct assessments at the word level. Our analysis shows that DAQE is more consistent with human judgment than TER-based QE datasets. Then, considering collecting such a golden dataset is expensive and labor-consuming, we further propose two automatic tag correcting strategies, namely tag refinement strategy and tree-based annotation strategy, which make the TER-based annotations more consistent with human judgment. We directly use the large-scale corrected TER-based dataset in the pre-training phase and achieve significant improvement on DAQE.

Our contributions can be summarized as follows: 1) We collect a new word-level QE dataset called DAQE that reflects human's direct assessments rather than the post-editing effort. We conduct detailed analyses and demonstrate two differences between DAQE and the previous TER-based dataset. 2) Considering data collection is labor-consuming, we also propose two automatic tag correcting strategies to make the TER-based artificial dataset more consistent with human judgment and then boost the performance by large-scale pre-training. 3) We conduct experiments on our collected DAQE dataset as well as the TER-based dataset MLQE-PE. The results of the automatic and human evaluation show that our approach not only achieves better performance but also demonstrates higher consistency with human judgment.

2 Data Collection and Analysis

2.1 Data Collection

To make our word-level DA annotations comparable to TER-generated ones, we directly take the source and MT texts from MLQE-PE (Fomicheva et al., 2020), the official dataset for the WMT20 QE shared task. It includes two language pairs that contain TER-generated annotations: English-German (En-De) and English-Chinese (En-Zh). The source texts are sampled from Wikipedia documents and the translations are obtained from the Transformer-

Dataset	Split	English-German				English-Chinese			
		samples	tokens	MT BAD tags	MT Gap BAD tags	samples	tokens	MT BAD tags	MT Gap BAD tags
MLQE-PE	train	7000	112342	31621 (28.15%)	5483 (4.59%)	7000	120015	65204 (54.33%)	10206 (8.04%)
	valid	1000	16160	4445 (27.51%)	716 (4.17%)	1000	17063	9022 (52.87%)	1157 (6.41%)
DAQE (ours)	train	7000	112342	10804 (9.62%)	640 (0.54%)	7000	120015	19952 (16.62%)	348 (0.27%)
	valid	1000	16160	1375 (8.51%)	30 (0.17%)	1000	17063	2459 (14.41%)	8 (0.04%)
	test	1000	16154	993 (6.15%)	28 (0.16%)	1000	17230	2784 (16.16%)	11 (0.06%)

Table 1: Statistics of TER-based MLQE-PE dataset and our proposed DAQE dataset.

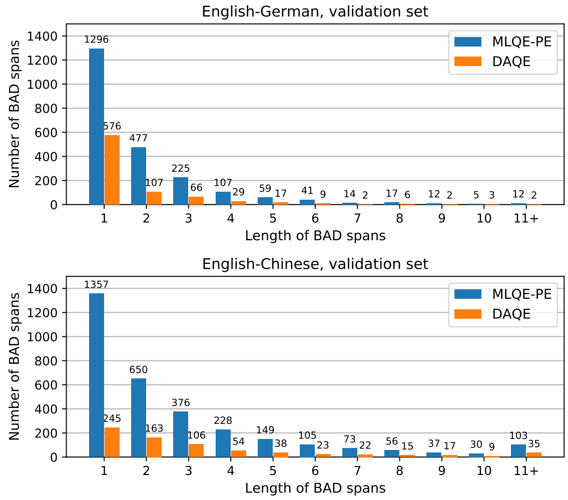


Figure 3: The length distribution of BAD spans.

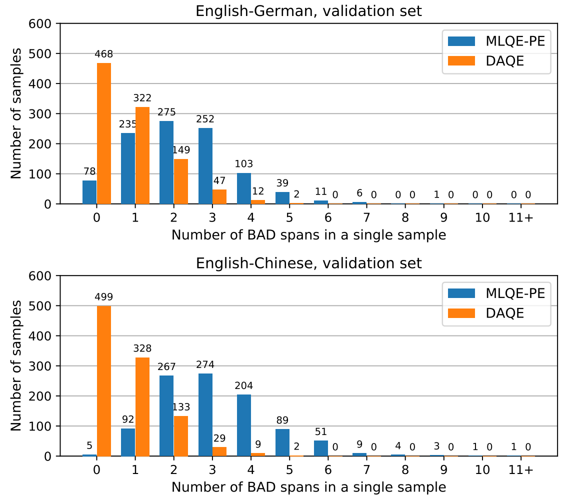


Figure 4: The distribution that reveals how many `BAD` spans in every single sample.

based neural machine translation (NMT) system (Vaswani et al., 2017).

To obtain the word-level DA annotations, we show human translators the source sentences with the corresponding MTs. Then we ask them to find words, phrases, clauses, or even the whole sentences that contain translation errors and annotate them as `BAD`, according to their professional knowledge. Note that although the PE sentences exist in MLQE-PE, the human annotators have no access to them, making the annotation process as fair and unbiased as possible. All of the annotated samples are cross-validated to ensure the accuracy rate above 95%.[3]

2.2 Statistics and Analysis

Overall Statistics. In Table 1, we show detailed statistics of MLQE-PE and DAQE. First, we see that the total number of `BAD` tags decreases heavily when human's DA replaces the TER-based annotations (from 28.15% to 9.62% for En-De, and from 54.33% to 16.62% for En-Zh). It indicates that the human's DA tends to annotate `OK` as long as the translation correctly expresses the meaning of the source sentence, but ignores the secondary issues like synonym substitutions and constituent reordering. Second, we find the number of `BAD` tags in the gap (indicating a few words are missing between two MT tokens) also greatly decreases. It's because that human's DA tends to regard the missing translations (i.e., the `BAD` gaps) and the translation errors as a whole but only annotate `BAD` tags on MT tokens[4].

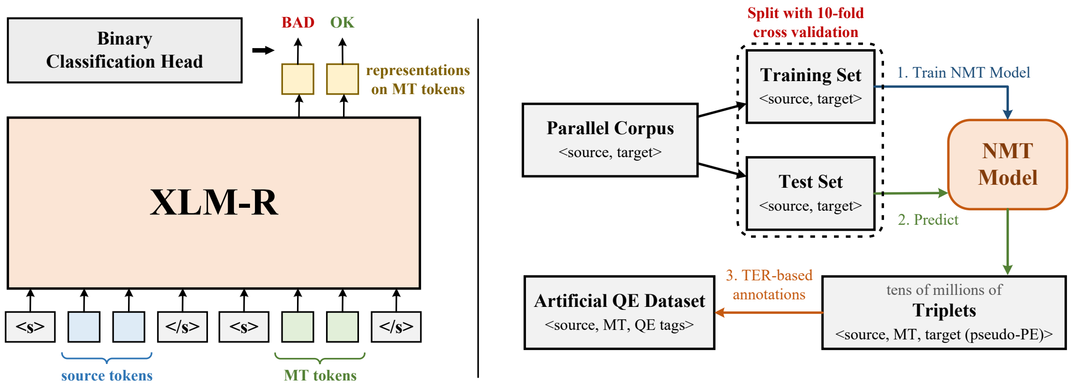
The Length of BAD Spans. We show the number of `BAD` spans[5] of different lengths in Figure 3. We can see that most `BAD` spans only contain a few tokens, showing the well-known long-tail distribution. For En-De, the long-tail distribution is sharper, where 70.5% of `BAD` spans are one-token spans. When comparing the TER-based annotations with the DA ones, we find that DA includes fewer `BAD` spans of each length, but the overall distribution is similar.

Unity of BAD Spans. To reveal the unity of the DA annotations, we group the samples according to the number of `BAD` spans in each single sample, and show the overall distribution. From Figure 4, we can find that the TER-based annotations follow the Gaussian distribution, where a large proportion of samples contain 2, 3, or even more `BAD`

[3]We provide more details in the Appendix E.

[4]As a result, we do not include the subtask of predicting gap tags in our experiments.

[5]Here, the `BAD` spans indicate the longest continuous tokens with `BAD` tags.



a) The overall architecture of our model. b) The construction steps of artificial QE dataset for pre-training.

Figure 5: The model architecture and the construction of artificial QE dataset.

spans, indicating the TER-based annotations are fragmented. However, our collected DA annotations are more unified, with only a small proportion of samples including more than 2 `BAD` spans. Besides, we find a large number of samples that are fully annotated as `OK` in the DA annotations. However, the number is extremely small for TER-based annotations (78 in English-German and 5 for English-Chinese). This shows a large proportion of `BAD` spans in TER-based annotations do not really destroy the semantic of translations and are thus regarded as `OK` by human's DA.

3 Approach

In this section, we will first introduce the backbone of the model and the construction of the TER-based artificial dataset for pre-training. Then, we propose two correcting strategies to make the TER-based artificial tags closer to the human judgment.

3.1 Model Architecture

Following (Ranasinghe et al., 2020; Lee, 2020; Moura et al., 2020; Ranasinghe et al., 2021), we select the XLM-RoBERTa (XLM-R) (Conneau et al., 2020) as the backbone of our model. XLM-R is a transformer-based masked language model pre-trained on large-scale multilingual corpus and demonstrates state-of-the-art performance on multiple cross-lingual downstream tasks. As shown in Figure 5a, we concatenate the source sentence and the MT sentence together to make an input sample: $\boldsymbol{x}_i = \texttt{<s>}w_1^{\text{src}}, \dots, w_m^{\text{src}}\texttt{</s><s>}w_1^{\text{mt}}, \dots, w_n^{\text{mt}}\texttt{</s>}$, where m is the length of the source sentence (src) and n is the length of the MT sentence (mt). `<s>` and `</s>` are two special tokens to annotate the start and the end of the sentence in XLM-R, respectively.

For the j-th token w_j^{mt} in the MT sentence, we take the corresponding representation from XLM-R for binary classification to determine whether w_j belongs to good translation (`OK`) or contains translation error (`BAD`) and use the binary classification loss to train the model:

$$s_{ij} = \sigma(\boldsymbol{w}^{\mathsf{T}}\text{XLM-R}_j(\boldsymbol{x}_i)) \tag{1}$$

$$\mathcal{L}_{ij} = -(y \cdot \log s_{ij} + (1-y) \cdot \log(1-s_{ij})) \tag{2}$$

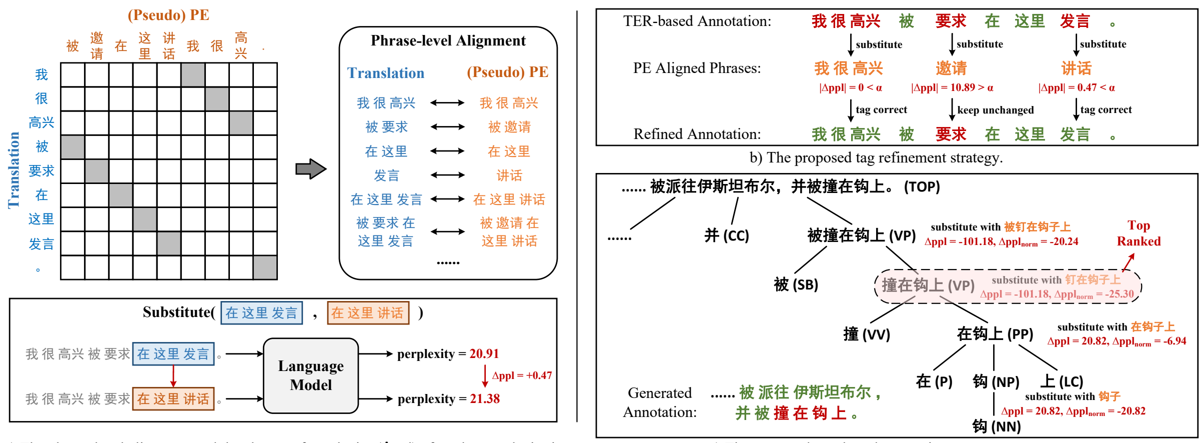
where $\text{XLM-R}_j(\boldsymbol{x}_i) \in \mathbb{R}^d$ (d is the hidden size of XLM-R) indicates the representation output by XLM-R corresponding to the token w_j^{mt}, σ is the sigmoid function, $\boldsymbol{w} \in \mathbb{R}^{d\times 1}$ is the linear layer for binary classification and y is the ground truth label.

3.2 Pre-training on Artificial QE Dataset

The translation knowledge contained in the parallel corpus of MT is very helpful for the QE task. As a result, many works use the parallel corpus for pre-training the model. As shown in Figure 5b, the parallel corpus is firstly split into the training and the test set. Then the NMT model is trained with the training split and is used to generate translations for all sentences in the test split. From this, a large number of triplets are obtained, each consisting of source, MT, and target sentences. Finally, the target sentence is regarded as the pseudo-PE from the MT sentence, and the TER toolkit is used to generate word-level `OK|BAD` tags based on the principle of minimal editing (shown in the bottom of Figure 1).

3.3 Tag Correcting Strategies

As we discussed before, the two issues of TER-based tags limit the performance improvement of pre-training when applied to the downstream DA



a) The phrase-level alignment and the change of perplexity (Δppl) after phrase substitution.

b) The proposed tag refinement strategy.

c) The proposed tree-based annotation strategy.

Figure 6: The proposed two tag correcting strategies: Tag Refinement strategy and Tree-based Annotation strategy.

task. In this section, we introduce two tag correcting strategies, namely tag refinement and tree-based annotation, that target these issues and make the TER-based artificial QE tags more consistent with human judgment.

Tag Refinement Strategy. In response to the first issue (i.e., wrong annotations due to the synonym substitution or constituent reordering), we propose the tag refinement strategy, which corrects the false `BAD` tags to `OK`. Specifically, as shown in Figure 6a, we first generate the alignment between the MT sentence and the reference sentence (i.e., the pseudo-PE) using FastAlign[6] (Dyer et al., 2013). Then we extract the phrase-to-phrase alignment through running the phrase extraction algorithm of NLTK[7] (Bird, 2006). Once the phrase-level alignment is prepared, we substitute each `BAD` span with the corresponding aligned spans in the pseudo-PE and use the language model to calculate the change of the perplexity Δppl after this substitution.

If $|\Delta ppl| < \alpha$, where α is a hyperparameter indicating the threshold, we regard that the substitution has little impact on the semantic and thus correct the `BAD` tags to `OK`. Otherwise, we regard the span does contain translation errors and keep the `BAD` tags unchanged (Figure 6b).

Tree-based Annotation Strategy. Human's DA tends to annotate the *smallest* constituent that causes fatal translation errors *as a whole* (e.g., the whole words, phrases, clauses, etc.). However, TER-based annotations are often fragmented, with the whole mistranslations being split into multiple `BAD` spans because some stopwords are aligned and labeled as `OK`. Besides, the `BAD` spans are often not well-formed in linguistics (e.g., two adjacent words but are from two different phrases).

To address this issue, we propose the constituent tree-based annotation strategy. It can be regarded as an enhanced version of the tag refinement strategy that gets rid of the TER-based annotation. As shown in Figure 6c, we first generate the constituent tree for the MT sentences. Each internal node (i.e., the non-leaf node) in the constituent tree represents a well-formed phrase such as noun phrase (NP), verb phrase (VP), prepositional phrase (PP), etc. For each node, we substitute it with the corresponding aligned phrase in the pseudo-PE. Then we still use the change of the perplexity Δppl to indicate whether the substitution of this phrase improves the fluency of the whole translation.

To only annotate the smallest constituents that exactly contain translation errors, we normalize Δppl by the number of words in the phrase and use this value to sort all internal nodes in the constituent tree: $\Delta ppl_{\text{norm}} = \frac{\Delta ppl}{r-l+1}$, where l and r indicates the left and right position of the phrase, respectively. The words of a constituent node are integrally labeled as `BAD` only if $\Delta ppl_{\text{norm}} < \beta$ as well as there is no overlap with nodes that are higher ranked. β is a hyperparameter indicating the threshold.

4 Experiments

Datasets. To verify the effectiveness of our proposed tag correcting strategies on word-level QE, we conduct experiments on both DAQE and MLQE-PE (Fomicheva et al., 2020) datasets. MLQE-PE is the official dataset used in the WMT20 QE shared task (Specia et al., 2020), and DAQE is our collected dataset with word-level DA

[6] https://github.com/clab/fast_align

[7] https://github.com/nltk/nltk/blob/develop/nltk/translate/phrase_based.py

Model	English-German (En-De)				English-Chinese (En-Zh)			
	MCC	F-OK	F-BAD	F-BAD-Span	MCC	F-OK	F-BAD	F-BAD-Span
Baselines								
FT on DAQE only	26.29	**95.08**	31.09	20.97	38.56	90.76	47.56	26.66
PT (TER-based)	9.52	34.62	13.54	3.09	15.17	36.66	31.53	2.40
+ FT on DAQE	24.82	94.65	29.82	18.52	39.09	**91.29**	47.04	25.93
Pre-training only with tag correcting strategies (ours)								
PT w/ Tag Refinement	10.12*	49.33	14.32	3.62	19.36*	53.16	34.10	3.79
PT w/ Tree-based Annotation	8.94	84.50	15.84	6.94	21.53*	59.21	35.54	6.32
Pre-training with tag correcting strategies + fine-tuning on DAQE (ours)								
PT w/ Tag Refinement + FT	27.54*	94.21	**35.25**	21.13	40.35*	90.88	49.33	25.60
PT w/ Tree-based Annotation + FT	**27.67***	94.44	32.41	**21.38**	**41.33***	91.22	**49.82**	**27.21**

Table 2: The word-level QE performance on the test set of DAQE for two language pairs, En-De and En-Zh. PT indicates pre-training and FT indicates fine-tuning. Results are all reported by $\times 100$. The numbers with * indicate the significant improvement over the corresponding baseline with $p < 0.05$ under t-test (Semenick, 1990).

annotations. Note that MLQE-PE and DAQE share the same source and MT sentences, thus they have exactly the same number of samples. We show the detailed statistics in Table 1. For the pre-training, we use the parallel dataset provided in the WMT20 QE shared task to generate the artificial QE dataset.

Baselines. To confirm the effectiveness of our proposed tag correcting strategies, we mainly select two baselines for comparison. In the one, we do not use the pre-training, but only fine-tune XLM-R on the training set of DAQE. In the other, we pre-train the model on the TER-based artificial QE dataset and then fine-tune it on the training set of DAQE.

Evaluation. Following WMT20 QE shared task (Specia et al., 2020), we use Matthews Correlation Coefficient (MCC) as the main metric and also provide the F1 score (F) for `OK`, `BAD` and `BAD` spans.[8]

4.1 Main Results

The results are shown in Table 2. We can observe that the TER-based pre-training only brings very limited performance gain or even degrade the performance when compared to the “FT on DAQE only” setting (-1.47 for En-De and +0.53 for En-Zh). It suggests that the inconsistency between TER-based and DA annotations leads to the limited effect of pre-training. However, when applying the tag correcting strategies to the pre-training dataset, the improvement is much more significant (+2.85 for En-De and +2.24 for En-Zh), indicating that the tag correcting strategies mitigate such inconsistency, improving the effect of pre-training. On the other hand, when only the pre-training is applied, the tag correcting strategies can also improve the performance. It shows our approach can also be applied to the unsupervised setting, where no human-annotated dataset is available for fine-tuning.

Tag Refinement v.s. Tree-based Annotation. When comparing two tag correcting strategies, we find the tree-based annotation strategy is generally superior to the tag refinement strategy, especially for En-Zh. The MCC improves from 19.36 to 21.53 under the *pre-training only* setting and improves from 40.35 to 41.33 under the *pre-training then fine-tuning* setting. This is probably because the tag refinement strategy still requires the TER-based annotation and fixes based on it, while the tree-based annotation strategy actively selects the well-formed constituents to apply phrase substitution and gets rid of the TER-based annotation.

Span-level Metric. Through the span-level metric (F-BAD-Span), we want to measure the unity and consistency of the model's prediction against human judgment. From Table 2, we find our models with tag correcting strategies also show higher F1 score on `BAD` spans (from 26.66 to 27.21 for En-Zh), while TER-based pre-training even do harm to this metric (from 26.66 to 25.93 for En-Zh). This phenomenon also confirms the aforementioned fragmented issue of TER-based annotations, and our tag correcting strategies, instead, improve the span-level metric by alleviating this issue.

4.2 Analysis

Comparison to results on MLQE-PE. To demonstrate the difference between the MLQE-PE (TER-generated tags) and our DAQE datasets, and analyze how the pre-training and fine-tuning influence the results on both datasets, we compare the performance of different models on MLQE-PE and DAQE respectively. The results for En-Zh are

[8] Please refer to Appendix A for implementation details.

Evaluate on → / Fine-tune on ↓	MLQE-PE			DAQE	
	MCC*	MCC	F-BAD	MCC	F-BAD
WMT20's best	59.28	-	-	-	-
No pre-training (fine-tuning only)					
MLQE-PE	**58.21**	**46.81**	**75.02**	22.49	34.34
DAQE	49.77	23.68	36.10	**45.76**	**53.77**
TER-based pre-training					
w/o fine-tune	56.51	33.58	73.85	11.38	27.41
MLQE-PE	**61.85**	**53.25**	**78.69**	21.93	33.75
DAQE	41.39	29.19	42.97	**47.34**	**55.43**
Pre-training with tag refinement					
w/o fine-tune	55.03	28.89	70.73	18.83	31.39
MLQE-PE	**61.35**	**48.24**	**77.17**	21.85	33.31
DAQE	39.56	25.06	67.40	**47.61**	**55.22**
Pre-training with tree-based annotation					
w/o fine-tune	55.21	26.79	68.11	20.98	32.84
MLQE-PE	**60.92**	**48.58**	**76.18**	22.34	34.13
DAQE	40.30	26.22	39.50	**48.14**	**56.02**

Table 3: Performance comparison for En-Zh with different fine-tuning and evaluation settings. Since the test labels of MLQE-PE are not publicly available, we report the results on the validation set of both datasets. MCC* indicates the MCC score considering both the target tokens and the target gaps.

shown in Table 3.

When comparing results in each group, we find that fine-tuning on the training set identical to the evaluation set is necessary for achieving high performance. Otherwise, fine-tuning provides marginal improvement (e.g., fine-tuning on MLQE-PE and evaluating on DAQE) or even degrades the performance (e.g., fine-tuning on DAQE and evaluating on MLQE-PE). This reveals the difference in data distribution between DAQE and MLQE-PE. Besides, we note that our best model on MLQE-PE outperforms WMT20's best model (61.85 v.s. 59.28) using the same MCC* metric, showing the strength of our model, even under the TER-based setting.

On the other hand, we compare the performance gain of different pre-training strategies. When evaluating on MLQE-PE, the TER-based pre-training brings higher performance gain (+6.44) than pre-training with two proposed tag correcting strategies (+1.43 and +1.77). While when evaluating on DAQE, the case is opposite, with the TER-based pre-training bringing lower performance gain (+1.58) than tag refinement (+1.85) and tree-based annotation (+2.38) strategies. In conclusion, the pre-training always brings performance gain, no matter evaluated on MLQE-PE or DAQE. However, the optimal strategy depends on the consistency between the pre-training dataset and the downstream evaluation task.

Models	En-De		En-Zh	
	Pea.	Spea.	Pea.	Spea.
Trained on sentence-level DA dataset				
WMT20's best	56.2	-	55.1	-
XLM-R Large	44.52	45.90	49.93	51.08
+ PT (HTER scores)	49.64	51.27	51.62	51.49
Derived from the prediction of word-level QE model				
FT on MLQE-PE	41.12	43.02	31.49	29.19
+ PT (TER-based)	38.88	42.22	33.08	31.41
FT on DAQE	**50.29**	**52.74**	42.33	43.48
+ PT (Tag Correcting)	50.07	51.04	**44.69**	**46.41**

Table 4: The Pearson's (Pea.) and Spearman's (Spea.) correlation ($\times$100) against the sentence-level DA scores on the validation set. HTER (Specia et al., 2020) indicates Human Translation Error Rate, a score derived from the TER-based tags.

Sentence-level DA Scores. Predicting sentence-level DA scores typically requires another model that trained on sentence-level QE task. However, with our word-level DA dataset, the sentence-level DA score can also be derived from word-level predictions. In this way, we can unify the DA predictions of word-level and sentence-level QE without the need of additional sentence-level DA dataset.

To show the performance of sentence-level DA score derived from the word-level DA model, we use the sentence-level DA scores in MLQE-PE as the gold scores and calculate the Pearson's correlation or Spearman's correlation between them and the model's predictions.

Table 4 illustrates the results. The first group gives the performance of sentence-level QE models that are trained on sentence-level DA datasets. Specially, we provide the best model[9] in the WMT20 QE shared task (sentence-level DA) and use them as a strong baseline.

In the second group, we obtain the sentence-level score by averaging the word-level scores: $s_i^{\text{sent}} = \frac{1}{|\boldsymbol{x}_i|}\sum_j s_{ij}$, where s_{ij} is the word-level score of the j-th token calculated by Equation 1. We can see the models trained on DAQE achieve higher sentence-level performance than those trained on MLQE-PE with a large margin (+9.17 for En-De and +11.61 for En-Zh). For En-De, Pearson's correlation (50.29) is even closer to WMT20's best model (56.2). Besides, our proposed tag correcting strategies can also improve the sentence-level performance for En-Zh (+2.36).

Human Evaluation. To evaluate and compare the models trained on TER-based tags and DA tags

[9] http://www.statmt.org/wmt20/quality-estimation-task_results.html

Scores	En-De		En-Zh	
	TER	DA	TER	DA
1 (terrible)	3	1	5	0
2 (bad)	36	16	34	6
3 (neutral)	34	20	29	21
4 (good)	26	61	24	59
5 (excellent)	1	2	8	14
Average score:	2.86	3.47	2.96	3.81
% DA $\geq$ TER:	89%		91%	

Table 5: The results of human evaluation. We select the best-performed model fine-tuned on MLQE-PE and DAQE respectively.

more objectively, human evaluation is conducted for both models. For En-Zh and En-De, we randomly select 100 samples (the source and MT sentences) from the validation set and use two models to predict word-level `OK` or `BAD` tags for them. Then, we ask human translators to give a score for each prediction, between 1 and 5, where 1 indicates the predicted tags are fully wrong, and 5 indicates the tags are fully correct.

Table 5 shows the results. We can see that the model trained on DA tags achieves higher human evaluation scores than that trained on TER-based tags on average. For about 90% of samples, the prediction of the DA model can outperform or tie with the prediction of TER-based model.

5 Related Work

Early approaches on QE, such as QuEst (Specia et al., 2013) and QuEst++ (Specia et al., 2015), mainly pay attention to the feature engineering. They aggregate various features and feed them to the machine learning algorithms for classification or regression. Kim et al. (2017) first propose the neural-based QE approach, called Predictor-Estimator. They first pre-train an RNN-based predictor on the large-scale parallel corpus that predicts the target word given its context and the source sentence. Then, they extract the features from the pre-trained predictor and use them to train the estimator for the QE task. This model achieves the best performance on the WMT17 QE shard task. After that, many variants of Predictor-Estimator are proposed (Fan et al., 2019; Moura et al., 2020; Cui et al., 2021). Among them, Bilingual Expert (Fan et al., 2019) replaces RNN with multi-layer transformers as the architecture of the predictor, and proposes the 4-dimension mismatching feature for each token. It achieves the best performance on WMT18 QE shared task. The Unbabel team also releases an open-source framework for QE, called OpenKiwi (Kepler et al., 2019), that implements the most popular QE models with configurable architecture.

Recently, with the development of pre-trained language models, many works select the cross-lingual language model XLM-RoBERTa (Conneau et al., 2020) as the backbone (Ranasinghe et al., 2020; Lee, 2020; Moura et al., 2020; Rubino and Sumita, 2020; Ranasinghe et al., 2021; Zhao et al., 2021). Many works also explore the joint learning or transfer learning of the multilingual QE task (i.e., on many language pairs) (Sun et al., 2020; Ranasinghe et al., 2020, 2021). Meanwhile, on the word-level QE, Fomicheva et al. (2021) propose a shared task with the new-collected dataset on explainable QE, aiming to provide word-level hints for sentence-level QE score. Freitag et al. (2021) also study multidimensional human evaluation for MT and collect a large-scale dataset.

The QE model can be applied to the Computer-Assisted Translation (CAT) system together with other models like translation suggestion (TS) or automatic post-edit (APE). Wang et al. (2020a) and Lee et al. (2021) use the QE model to identify which parts of the machine translations need to be correct, and the TS (Yang et al., 2021) also needs the QE model to determine error spans before giving translation suggestions.

6 Conclusion

In this paper, we focus on the task of word-level QE in machine translation and target the inconsistency issues between the TER-based QE dataset and human judgment. We for the first time collect a word-level QE dataset called DAQE that reflects human's direct assessments. Besides, we propose two tag correcting strategies that correct the TER-based artificial QE tags in the pre-training phase and further improve the performance. We conduct thorough experiments and analyses, demonstrating the necessity of our proposed dataset and the effectiveness of our proposed approaches. Our future directions include improving the performance of phrase-level alignment, introducing phrase-level semantic matching, and applying data augmentation[10]. We hope our work will provide a new perspective for future researches on quality estimation.

[10] We provide case studies and discuss the current limitations and potential strategies in the appendix.

Broader Impacts

Quality estimation often serves as a post-processing module in recent commercial machine translation systems. It can be used to indicate the overall translation quality or detect the specific translation errors in the sentences. This work focuses on the direct assessment task, training the model to fit the human judgment at the word level. To do this, we collect a new QE dataset and propose tag correcting strategies to force the TER-based artificial dataset used in the pre-training phase closer to human judgment. When applying our approach, the users should pay special attention to the following: a) The data source of DAQE is Wikipedia, so our model should perform well on a similar domain but may perform poorly on other irrelevant domains. b) Since our approach is still data-driven, the data (as well as the pre-training parallel dataset) should be ethical and unbiased, or unexpected problems may arise. c) The proposed tag correcting strategies work well on En-De and En-Zh, but do not necessarily applicable to other language pairs since the characteristics among target languages are different. d) Since the system is neural-based, the interpretability is limited. It can still mistakenly annotate some forbidden or sensitive words to `OK` and cause unexpected accidents.

References

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

Wanxiang Che, Zhenghua Li, and Ting Liu. 2010. Ltp: A chinese language technology platform. In *Coling 2010: Demonstrations*, pages 13–16.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Qu Cui, Shujian Huang, Jiahuan Li, Xiang Geng, Zaixiang Zheng, Guoping Huang, and Jiajun Chen. 2021. Directqe: Direct pretraining for machine translation quality estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12719–12727.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Kai Fan, Jiayi Wang, Bo Li, Fengming Zhou, Boxing Chen, and Luo Si. 2019. "bilingual expert" can find translation errors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6367–6374.

Marina Fomicheva, Piyawat Lertvittayakumjorn, Wei Zhao, Steffen Eger, and Yang Gao. 2021. The eval4nlp shared task on explainable quality estimation: Overview and results. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 165–178.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André FT Martins. 2020. Mlqe-pe: A multilingual quality estimation and post-editing dataset. *arXiv preprint arXiv:2010.04480*.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *arXiv preprint arXiv:2104.14478*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André FT Martins. 2019. Openkiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(1):1–22.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Dongjun Lee. 2020. Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1024–1028.

Dongjun Lee, Junhyeong Ahn, Heesoo Park, and Jaemin Jo. 2021. Intellicat: Intelligent machine translation post-editing with quality estimation and translation suggestion. *arXiv preprint arXiv:2105.12172*.

Joao Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André FT Martins. 2020. Ist-unbabel participation in the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1029–1036.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. Transquest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2021. An exploratory analysis of multilingual word-level quality estimation with cross-lingual transformers. *arXiv preprint arXiv:2106.00143*.

Raphael Rubino and Eiichiro Sumita. 2020. Intermediate self-supervised learning for machine translation quality estimation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4355–4360.

Doug Semenick. 1990. Tests and measurements: The t-test. *Strength & Conditioning Journal*, 12(1):36–37.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. 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*, pages 223–231.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120.

Lucia Specia, Kashif Shah, José GC De Souza, and Trevor Cohn. 2013. Quest-a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84.

Shuo Sun, Marina Fomicheva, Frédéric Blain, Vishrav Chaudhary, Ahmed El-Kishky, Adithya Renduchintala, Francisco Guzmán, and Lucia Specia. 2020. An exploratory study on multilingual quality estimation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 366–377.

Yi-Lin Tuan, Ahmed El-Kishky, Adithya Renduchintala, Vishrav Chaudhary, Francisco Guzmán, and Lucia Specia. 2021. Quality estimation without human-labeled data. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 619–625, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ke Wang, Jiayi Wang, Niyu Ge, Yangbin Shi, Yu Zhao, and Kai Fan. 2020a. Computer assisted translation with neural quality estimation and auotmatic post-editing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2175–2186.

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

Shufan Wang, Laure Thompson, and Mohit Iyyer. 2021. Phrase-bert: Improved phrase embeddings from bert with an application to corpus exploration. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10837–10851.

Zhen Yang, Yingxue Zhang, Ernan Li, Fandong Meng, and Jie Zhou. 2021. Wets: A benchmark for translation suggestion. *arXiv preprint arXiv:2110.05151*.

Mingjun Zhao, Haijiang Wu, Di Niu, Zixuan Wang, and Xiaoli Wang. 2021. Verdi: Quality estimation and error detection for bilingual corpora. In *Proceedings of the Web Conference 2021*, pages 3023–3031.

A Implementation Details

Our implementation of QE model is based on an open-source framework, OpenKiwi[11] (Kepler et al., 2019). We use the large-sized XLM-R model and obtain it from hugging-face's library[12]. We use the KenLM[13] (Heafield, 2011) to train the language model on all target sentences in the parallel corpus and calculate the perplexity of the given sentence. For the tree-based annotation strategy, we obtain the constituent tree through LTP[14] (Che et al., 2010) for Chinese and through Stanza[15] (Qi et al., 2020) for German. We set α to 1.0 and β to -3.0 in our tag correcting strategies based on the case studies and empirical judgment. In the preprocessing phase, we filter out parallel samples that are too long or too short, and only reserve sentences with 10-100 tokens.

We pre-train the model on 8 NVIDIA Tesla V100 (32GB) GPUs for two epochs, with the batch size set to 8 for each GPU. Then we fine-tune the model on a single NVIDIA Tesla V100 (32GB) GPU for up to 10 epochs, with the batch size set to 8 as well. Early stopping is used in the fine-tuning phase, with the patience set to 20. We evaluate the model every 10% steps in one epoch. The pre-training often takes more than 15 hours and the fine-tuning takes 1 or 2 hours. We use Adam (Kingma and Ba, 2014) to optimize the model with the learning rate set to 5e-6 in both the pre-training and fine-tuning phases. For all hyperparameters in our experiments, we manually tune them on the validation set of DAQE.

B Main Results on the Validation Set

In Table 6, we also report the main results on the validation set of DAQE.

C Case Study

In Figure 7, we show some cases from the validation set of English-Chinese language pair. From the examples, we can see that the TER-based model (noted as PE Effort Prediction) often annotates wrong `BAD` spans and is far from human judgment. For the first example, the MT sentence correctly reflects the meaning of the source sentence, and the PE is just a paraphrase of the MT sentence. Our DA model correctly annotates all words as `OK`, while TER-based one still annotates many `BAD` words. For the second example, the key issue is the translation of "unifies" in Chinese. Though "统一" is the direct translation of "unifies" in Chinese, it can not express the meaning of winning two titles in Chinese context. And our DA model precisely annotated the "统一 了" in the MT sentence as `BAD`. For the third example, the MT model fails to translate the "parsley" and the "sumac" to "欧芹" and "盐肤木" in Chinese, since they are very rare words. While the TER-based model mistakenly predicts long `BAD` spans, our DA model precisely identities both mistranslation parts in the MT sentence.

D Limitation and Discussion

We analyze some samples that are corrected by our tag correcting strategies and find a few bad cases. These are mainly because of the following: 1) There is noise from the parallel corpus (i.e., the source sentence and the target sentence are not well aligned). 2) The alignment generated by FastAlign contains unexpected errors, making some entries in the phrase-level alignments are missing or misaligned. 3) The scores given by KenLM (through the change of the perplexity after the phrase substitution) are sometimes not consistent with human judgment.

We also propose some possible solutions in response to the above problems as our future exploration direction. For the noise in the parallel corpus, we can use parallel corpus filtering methods that filter out samples with low confidence. We can also apply the data augmentation methods that expand the corpus based on the clean parallel corpus. For the errors by FastAlign, we may use a more accurate alignment model. For the scoring, we may introduce the neural-based phrase-level semantic matching model (e.g., Phrase-BERT (Wang et al., 2021)) instead of the KenLM.

E Details about Data Collection

The number of translators are 5 for En-Zh and 6 for En-De. They are all graduated students that major in the translation (with the professional ability on the corresponding source and target languages).

For each sample, we randomly distribute it to two annotators. If they have annotation conflicts, we will ask another annotator as the referee to judge

[11] https://github.com/Unbabel/OpenKiwi

[12] https://huggingface.co/xlm-roberta-large

[13] https://kheafield.com/code/kenlm.tar.gz

[14] http://ltp.ai/index.html

[15] https://stanfordnlp.github.io/stanza/index.html

Model	English-German (En-De)				English-Chinese (En-Zh)			
	MCC	F-OK	F-BAD	F-BAD-Span	MCC	F-OK	F-BAD	F-BAD-Span
Baselines								
FT on DAQE only	34.69	94.28	40.38	28.65	45.76	91.96	53.77	**29.84**
PT (TER-based)	13.13	37.30	18.80	4.72	11.38	25.91	27.41	2.16
+ FT on DAQE	35.02	94.00	40.86	26.68	47.34	91.30	55.43	28.53
With tag correcting strategies (ours)								
PT w/ Tag Refinement	13.26	52.43	19.78	6.42	18.83	53.29	31.39	3.48
+ FT on DAQE	37.70	94.08	43.32	30.83	47.61	**92.39**	55.22	28.33
PT w/ Tree-based Annotation	13.92	84.79	22.75	9.64	20.98	59.32	32.84	6.53
+ FT on DAQE	37.03	**94.46**	42.54	31.21	48.14	91.88	56.02	28.17
PT w/ Both	13.12	39.68	18.94	5.26	21.39	56.76	32.74	5.72
+ FT on DAQE	**38.90**	94.44	**44.35**	**32.21**	**48.71**	90.74	**56.47**	25.51

Table 6: The word-level QE performance on the validation set of DAQE for two language pairs, En-De and En-Zh. PT indicates pre-training and FT indicates fine-tuning.

Source: To win, a wrestler must strip their opponent's tuxedo off.
MT: 要 想 获胜 , 摔跤 运动员 必须 把 对手 的 礼服 脱下来 .
MT Back: To win, the wrestler had to take his opponent's dress off.
PE: 要 赢得 胜利 ， 摔跤 运动员 必须 脱掉 对手 的 燕尾服 。
PE Back: To win the victory, the wrestler had to remove his opponent's tuxedo.

PE Effort Prediction: **要 想 获胜 , 摔跤 运动员 必须 把 对手 的 礼服 脱下来 .**
DA Prediction: **要 想 获胜 , 摔跤 运动员 必须 把 对手 的 礼服 脱下来 .**

Source: April 28 Juan Díaz unifies the WBA and WBO Lightweight titles after defeating Acelino Freitas.
MT: 4 月 28 日 , 胡安 · 迪亚斯 在 击败 阿 切利 诺 · 弗雷 塔斯 后 统一 了 WBA 和 WBO 轻量级 冠军 .
MT Back: On April 28, Juan Díaz Unified the WBA and WBO lightweight titles after defeating Acelino Freitas.
PE: 4 月 28 日 ， Juan Díaz 在 击败 Acelino Freitas 之后 ， 将 W 世界 拳击 协会 和 世界 拳击 组织 的 轻量级 冠军 揽于 一身 。
PE Back: On April 28, Juan Díaz won both the WBA and WBO lightweight titles after defeating Acelino Freitas.

PE Effort Prediction: **4 月 28 日 , 胡安 · 迪亚斯 在 击败 阿 切利 诺 · 弗雷 塔斯 后 统一 了 WBA 和 WBO 轻量级 冠军 .**
DA Prediction: **4 月 28 日 , 胡安 · 迪亚斯 在 击败 阿 切利 诺 · 弗雷 塔斯 后 统一 了 WBA 和 WBO 轻量级 冠军 .**

Source: Fattoush is a combination of toasted bread pieces and parsley with chopped cucumbers, radishes, tomatoes and flavored by sumac.
MT: 法杜什是 烤面包片 和 帕斯 莱 与 切碎 的 黄瓜 、 萝卜 、 西红柿 、 和 洋葱 以及 香味 的 消耗品 的 组合 。
MT Back: Fadush is a combination of toast and pasai with chopped cucumbers, radishes, tomatoes and onions and scented consumables.
PE: Fattoush 是 烤面包片 和 欧芹 与 切碎 的 黄瓜 ， 萝卜 ， 西红柿 和 葱 的 组合 ， 并 以 盐肤木 调味 。
PE Back: Fattoush is a combination of toast and parsley with chopped cucumbers, radishes, tomatoes and scallions, seasoned with rhus salt.

PE Effort Prediction: **法杜什是 烤面包片 和 帕斯 莱 与 切碎 的 黄瓜 、 萝卜 、 西红柿 、 和 洋葱 以及 香味 的 消耗品 的 组合 。**
DA Prediction: **法杜什是 烤面包片 和 帕斯 莱 与 切碎 的 黄瓜 、 萝卜 、 西红柿 、 和 洋葱 以及 香味 的 消耗品 的 组合 。**

Figure 7: Examples of word-level QE from the validation set of English-Chinese language pair.

which annotation is better.

The annotator is provided only the source sentence and its corresponding translation (but without the context or passage which the source sentence is taken from). For En-Zh, the translations are tokenized (as they are in MLQE-PE).

The annotation and distribution of samples are automatically conducted through the annotation system. After all samples are annotated, we ask another translator (1 for En-Zh and 1 for En-De, and they do not participant in the annotation process), sampling a small proportion (400 samples) of the full annotated dataset and ensure the accuracy is above 95%.

For the annotation protocal, we ask human translators to find words, phrases, clauses or even the whole sentences that contain translation error in MT sentences, and annotate them as `BAD` tags. Here, the translation error means the translation distorts the meaning of the source sentence, but excluding minor mismatches such as synonyms and punctuation. Meanwhile, if the translation does not conform to the grammar of the target language, they should also find them and annotate as `BAD`. Besides, we also highlight the coherence of DA annotations, that is, human translators should annotate the smallest syntactic components that lead to translation errors as a whole.