Self-Supervised Quality Estimation for Machine Translation

Yuanhang Zheng[1,3,4], **Zhixing Tan**[1,3,4], **Meng Zhang**[6], **Mieradilijiang Maimaiti**[1,3,4], **Huanbo Luan**[1,3,4], **Maosong Sun**[1,3,4,5], **Qun Liu**[6] **and Yang Liu**[*1,2,3,4,5]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Institute for AI Industry Research, Tsinghua University, Beijing, China
[3]Institute for Artificial Intelligence, Tsinghua University, Beijing, China
[4]Beijing National Research Center for Information Science and Technology
[5]Beijing Academy of Artificial Intelligence
[6]Huawei Noah's Ark Lab

Abstract

Quality estimation (QE) of machine translation (MT) aims to evaluate the quality of machine-translated sentences without references and is important in practical applications of MT. Training QE models require massive parallel data with hand-crafted quality annotations, which are time-consuming and labor-intensive to obtain. To address the issue of the absence of annotated training data, previous studies attempt to develop unsupervised QE methods. However, very few of them can be applied to both sentence- and word-level QE tasks, and they may suffer from noises in the synthetic data. To reduce the negative impact of noises, we propose a self-supervised method for both sentence- and word-level QE, which performs quality estimation by recovering the masked target words. Experimental results show that our method outperforms previous unsupervised methods on several QE tasks in different language pairs and domains.[1]

1 Introduction

In recent years, neural approaches (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015; Vaswani et al., 2017) have significantly improved the quality of machine translation (MT). Despite their apparent success, neural machine translation (NMT) systems still inevitably generate erroneous translations in real-world scenarios (Bentivogli et al., 2016; Castilho et al., 2017), especially for low-resource language pairs. Therefore, the evaluation of translation quality plays an important role in some applications of MT. For example, in computer-assisted translation (CAT) (Barrachina et al., 2009), the evaluation of translation quality can significantly reduce human efforts for post-editing (Specia, 2011).

Quality estimation (QE) of MT aims to evaluate the quality of the outputs of an MT system without references. Training QE models often requires massive parallel data, which are composed of authentic source sentences and machine-translated target sentences with quality annotations produced by manual evaluation or human post-editing (Moura et al., 2020; Hu et al., 2020; Ranasinghe et al., 2020). As obtaining such annotated data is time-consuming and labor-intensive in practice, unsupervised QE has received increasing attention (Popović, 2012; Etchegoyhen et al., 2018; Zhang et al., 2020; Zhou et al., 2020; Fomicheva et al., 2020; Tuan et al., 2021).

Most of the aforementioned methods use various features to conduct unsupervised QE (Popović, 2012; Etchegoyhen et al., 2018; Zhang et al., 2020; Zhou et al., 2020; Fomicheva et al., 2020). These methods are simple and effective but limited to sentence-level tasks. Compared with sentence-level QE, word-level QE can provide more fine-grained quality information (Fan et al., 2019), and thus it can better assist post-editing in CAT when combined with sentence-level QE. Recently, Tuan et al. (2021) use synthetic data to train unsupervised QE models, which can be applied for both sentence- and word-level tasks. Specifically, they construct synthetic target sentences using MT models or masked language models (MLMs) and generate quality annotations by comparing the synthetic target sentences with the references using the TER tool (Snover et al., 2005).

However, the method proposed by Tuan et al. (2021) still has two major weaknesses. First, synthetic data contain biased noise and may negatively affect the model performance. On the one hand, the differences between MT outputs and references are usually larger than the differences between MT outputs and their post-editions (Snover et al., 2005), and thus more errors will be annotated in the synthetic data. On the other hand, sentences that are

*Corresponding author
[1]Code can be found at https://github.com/THUNLP-MT/SelfSupervisedQE.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3322–3334
November 7–11, 2021.

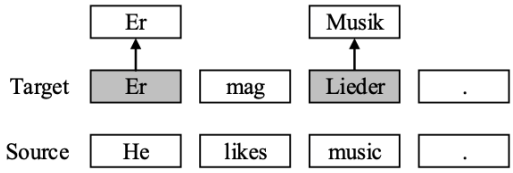


Figure 1: Overview of our self-supervised QE method. Our method performs quality estimation by checking whether the masked target words can be successfully recovered using the source sentence and the observed target words. Masked words are highlighted by shading.

Source	He	likes	Music	.
Target	Er	mag	*Lieder*	.
Post-edition	Er	mag	Musik	.
Sentence-level QE	0.25			
Word-level QE	OK	OK	BAD	OK

Table 1: Example of QE data for English-German translation. Erroneous words in MT are highlighted in italic.

rewritten by MLMs often contain more catastrophic errors, which rarely appear in machine-translated sentences (Tuan et al., 2021). Second, the training process of this method is complex since it requires extra models to generate synthetic data.

In this work, we propose a self-supervised QE method to overcome the aforementioned weaknesses. The basic idea is to mask some target words in the machine-translated sentence and use the source sentence and the observed target words to recover the masked target words. Intuitively, a target word is correct if it can be recovered according to its surrounding context. For example, in Figure 1, since the masked target word "Er" can be successfully recovered but another masked target word "Lieder" is not identical to the recovered word "Musik", we identify "Er" as correct and "Lieder" as erroneous. Based on this intuition, our method estimates the translation quality of the target words by checking whether they can be correctly recovered. Finally, we obtain the sentence-level quality score by summarizing the word-level predictions. Obviously, our method is not affected by the noise and is easier to train, since it involves no synthetic data. Experimental results show that our self-supervised method outperforms previous unsupervised methods.

2 Quality Estimation for Machine Translation

Quality estimation for machine translation aims to evaluate the quality of machine-translated sentences without using references. Currently, there are different types of QE tasks, including sentence-, word-, phrase- and document-level QE. In this work, we mainly focus on sentence- and word-level QE.

Generally, both sentence- and word-level quality annotations are generated by comparing the machine-translated target sentences with their post-editions using the TER tool (Snover et al., 2005). For word-level annotations, each target word is annotated with "OK" or "BAD", where "OK" denotes correct words and "BAD" denotes erroneous words. For sentence-level annotations, target sentences are annotated with Human Translation Error Rate (HTER) scores, which measure the percentage of human edits to correct MT outputs:

$$\text{HTER} = \frac{\text{\# of edits}}{\text{\# of words in the post-edition}}. \quad (1)$$

According to the equation above, sentence-level quality scores are calculated based on the word-level errors in the target sentences. In other words, HTER scores can be approximately regarded as a summary of word-level quality tags. Table 1 shows an example of QE data.

3 Self-Supervised Quality Estimation

Our self-supervised QE method is implemented based on the architecture of MLM (Devlin et al., 2019) (Section 3.1). We train the model to recover the masked target words in the authentic parallel corpora and estimate the translation quality by recovering the masked target words in the target sentence (Section 3.2). Besides, Monte Carlo (MC) Dropout (Gal and Ghahramani, 2016) is utilized to better calculate quality scores (Section 3.3).

3.1 Model Architecture

As shown in Figure 2, our self-supervised QE model is built on top of the masked language model (Devlin et al., 2019). We use the concatenation of a source sentence and a partially masked target sentence as the input sequence and then use a Transformer encoder to recover the masked tokens. Formally, for any parallel sentence pair $\langle \mathbf{x}, \mathbf{y} \rangle$, we randomly divide $\mathbf{y}$ into two parts $\mathbf{y}_m$ and $\mathbf{y}_o$ and mask all tokens in $\mathbf{y}_m$. Then, we concatenate $\mathbf{x}$ and the partially masked version of $\mathbf{y}$ as the input sequence. Suppose the length of the target sentence is

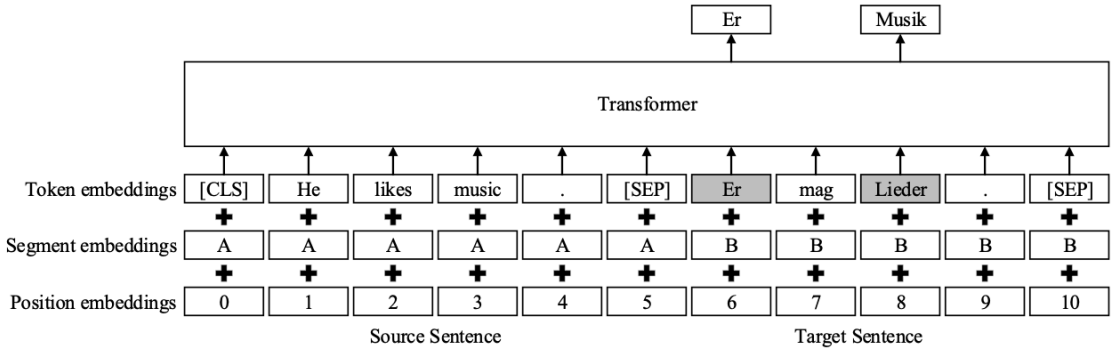


Figure 2: The model architecture of our self-supervised QE method. Masked words are highlighted by shading.

T: $\mathbf{y} = y_1, \ldots, y_t, \ldots, y_T$. If the t-th target token $y_t \in \mathbf{y}_m$ is masked, we use the model with parameter $\boldsymbol{\theta}$ to calculate the probability of y_t conditioned on $\mathbf{x}$ and $\mathbf{y}_o$ (i.e., $P(y_t|\mathbf{x}, \mathbf{y}_o; \boldsymbol{\theta})$).

Similar to Devlin et al. (2019), we mask 15% of the tokens in the target sentence. However, since the vocabulary of BERT is built with WordPiece (Wu et al., 2016), words in the input sequence may be divided into multiple subwords. Therefore, when a subword of a word with multiple subwords is masked, the model may easily recover the masked subword according to the remaining subwords without leveraging the source sentence. This is undesirable because the source sentence should play an important role in determining whether the token is correctly translated. To address this problem, we adopt a masking strategy called Whole Word Masking (WWM) (Cui et al., 2019), preventing the model from recovering a masked subword only using the remaining subwords.

3.2 Training and Inference

As shown in Figure 3(a), our model is trained to recover the masked tokens in the target side of the authentic sentence pairs. Formally, given an unlabeled training dataset $\mathcal{D} = \{\langle \mathbf{x}^{(s)}, \mathbf{y}^{(s)} \rangle\}_{s=1}^{S}$ which consists of authentic sentence pairs, we divide each target sentence $\mathbf{y}^{(s)}$ in $\mathcal{D}$ into the masked part $\mathbf{y}_m^{(s)}$ and the observed part $\mathbf{y}_o^{(s)}$. We train the model on $\mathcal{D}$ to minimize the negative log-likelihood loss on the masked target tokens:

$$\begin{aligned}\mathcal{L}(\mathcal{D}, \boldsymbol{\theta}) &= -\sum_{s=1}^{S} \log P(\mathbf{y}_m|\mathbf{x}^{(s)}, \mathbf{y}_o^{(s)}; \boldsymbol{\theta}) \\ &= -\sum_{s=1}^{S} \sum_{y_t \in \mathbf{y}_m^{(s)}} \log P(y_t|\mathbf{x}^{(s)}, \mathbf{y}_o^{(s)}; \boldsymbol{\theta}).\end{aligned} \tag{2}$$

During the training process, the model $\boldsymbol{\theta}$ learns to recover the masked target tokens in the authentic parallel corpora. After the training process, we use the model to perform quality estimation by checking whether the masked target tokens can be successfully recovered. Specifically, as shown in Figure 3(b), for each masked token, we use the model to calculate the probability of successful recovery conditioned on the source sentence and the observed target tokens. Obviously, the token is difficult to recover if the probability is low. In this case, we consider the token is erroneous. Otherwise, the token tends to be correct.

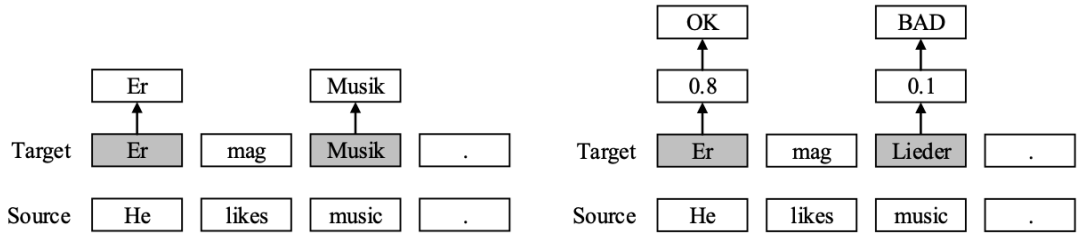
Formally, suppose we have a sentence pair $\langle \mathbf{x}, \hat{\mathbf{y}} \rangle$ which consists of an authentic source sentence $\mathbf{x}$ and a machine-translated target sentence $\hat{\mathbf{y}}$. When estimating the translation quality of the t-th token $\hat{y}_t$ in $\hat{\mathbf{y}}$, our method randomly divides the target sequence $\hat{\mathbf{y}}$ into the observed part $\hat{\mathbf{y}}_o$ and the masked part $\hat{\mathbf{y}}_m$ such that $\hat{y}_t \in \hat{\mathbf{y}}_m$. Then, we use the model to calculate the conditional probability $P(\hat{y}_t|\mathbf{x}, \hat{\mathbf{y}}_o; \boldsymbol{\theta})$, which can be used as its quality score:

$$score(\hat{y}_t) = P(\hat{y}_t|\mathbf{x}, \hat{\mathbf{y}}_o; \boldsymbol{\theta}). \tag{3}$$

As mentioned in Section 3.1, some of the input words may contain multiple subwords. In this case, we use $\hat{y}_t$ to denote a subword in the target sequence and $\hat{\mathbf{w}}$ to denote the word which $\hat{y}_t$ belongs to. We calculate the quality score of a target word with multiple subwords by simply averaging the quality scores of its subwords:

$$score(\hat{\mathbf{w}}) = \frac{1}{|\hat{\mathbf{w}}|} \sum_{\hat{y}_t \in \hat{\mathbf{w}}} score(\hat{y}_t), \tag{4}$$

where $|\hat{\mathbf{w}}|$ denotes the number of subwords in $\hat{\mathbf{w}}$.



(a) Training (b) Inference

Figure 3: Training and inference processes of our self-supervised QE method. Masked words are highlighted by shading. (a) During training, the model is trained to recover the masked target words in authentic parallel sentence pairs. (b) During inference, the model performs quality estimation according to the probability that the masked target words are successfully recovered.

If a threshold $\tau \in (0, 1)$ is given, a real-valued quality score can be mapped to a quality tag:

$$tag(\hat{\mathbf{w}}) = \begin{cases} \text{OK} & score(\hat{\mathbf{w}}) \geq \tau, \\ \text{BAD} & score(\hat{\mathbf{w}}) < \tau. \end{cases} \quad (5)$$

Finally, we calculate the sentence-level quality score by averaging the quality scores over all target words:

$$score(\hat{\mathbf{y}}) = -\frac{1}{|\hat{\mathbf{y}}|} \sum_{\hat{\mathbf{w}} \in \hat{\mathbf{y}}} score(\hat{\mathbf{w}}), \quad (6)$$

where $|\hat{\mathbf{y}}|$ denotes the number of words in $\hat{\mathbf{y}}$. Note that we add a negative sign to the equation above since HTER scores are negatively correlated with translation quality.

3.3 Calculating Quality Scores with Monte Carlo Dropout

In this work, we also utilize Monte Carlo (MC) Dropout (Gal and Ghahramani, 2016), which is proven conducive to the performance of unsupervised QE models (Fomicheva et al., 2020). Instead of directly calculating token-level quality scores using Eq. (3), we sample multiple models by perturbing the original model parameters using dropout (Srivastava et al., 2014) and use these models to calculate the expectation of conditional probabilities as the quality scores.

Specifically, in our method, each time we can only obtain the probability of the masked target words. Therefore, if we need N samples of probability for each target token, we sample $N' > N$ different models and conduct N' different estimations using these models, making each target word masked exactly N times among all N' estimations. Thus, we obtain N samples for each target token and then calculate the quality score by averaging these samples. For the details about this process, please refer to Appendix A.1.

4 Experiments

4.1 Setup

Data and Preprocessing

We mainly conducted experiments on the WMT 2019 QE tasks, which consist of tasks in two different language pairs (En-De and En-Ru). Both tasks are in the IT domain. Since our experiments were conducted in an unsupervised setting, we used parallel corpora without quality annotations as training data[2]. Specifically, for En-De, we used in-domain parallel data from various sources, including the training data from the WMT 2016 IT domain translation task, the WMT 2017 QE task, and the WMT 2018 APE task, as well as the Openoffice and KDE4 corpora available in OPUS[3] (Tiedemann, 2012). For En-Ru, we used the in-domain parallel data collected by OPUS, including ada83, GNOME, KDE4, OpenOffice, PHP and Ubuntu.

To further validate our method's performance in different domains, we also conducted experiments on the WMT 2018 En-Lv QE task, which is in the biomedical domain. We used the EMEA corpus (which is also available in OPUS) as training data.

Sentences were tokenized and truecased using the scripts provided by Moses (Koehn et al., 2007). We also deduplicated the sentences in the training datasets. Table 2 shows the statistics of these datasets.

[2]Although some of the training data have quality annotations, we did not use these annotations in the experiments.

[3]https://opus.nlpl.eu/

Year	Language Pair	Domain	System	Train	Dev	Test
2018	En-Lv	Biomedical	SMT	313K	1.00K	1.32K
			NMT		1.00K	1.45K
2019	En-De	IT	NMT	365K	1.00K	1.02K
	En-Ru	IT	NMT	217K	1.00K	1.02K

Table 2: Statistics of the training, development and test datasets in our experiments.

Baselines

We mainly compared our method with SyntheticQE (Tuan et al., 2021), which uses synthetic data to train unsupervised QE models for both sentence- and word-level tasks. This baseline has three different variants:

1. SyntheticQE-MT: The target side of the synthetic data is produced using MT models.

2. SyntheticQE-MLM: The target side of the synthetic data is produced using MLMs.

3. SyntheticQE-MT+MLM: An ensemble of the aforementioned two models.

To further validate the performance of our method, we also compared our method with the following unsupervised sentence-level baseline methods:

1. uMQE (Etchegoyhen et al., 2018): A method based on lexical translation tables and statistical language models.

2. BERTScore (Zhang et al., 2020): A method based on similarity scores of contextual BERT embeddings.

3. BERTScore++ (Zhou et al., 2020): A variant of BERTScore (Zhang et al., 2020), which also uses word alignments and MLMs.

4. NMT-QE (Fomicheva et al., 2020): A method based on NMT models and uncertainty quantification.

Evaluation

We evaluated the performances of our method and the baselines using the standard metrics of the WMT QE shared tasks. Specifically, we used Pearson's correlation metric for sentence-level tasks and the multiplication of F1-scores for "OK" and "BAD" classes for word-level tasks.

Implementation Details

We implemented our method on top of the Transformers library[4] (Wolf et al., 2020). We trained our model by fine-tuning the multilingual BERT (Devlin et al., 2019). We used the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ to optimize model parameters. During training, we set the batch size to 128, the maximum sequence length to 256, the number of training steps to 100,000, the learning rate to $5{\times}10^{-5}$ and the dropout rate to 0.1. We evaluated our model every 1,000 steps and chose the model with the best performance on the development set for inference. For the MC Dropout process, we used the same dropout rate as during training and set N to 6. Since each prediction masks about 15% of the words, we set $N' = N/15\% = 40$. We tuned the threshold τ on the development set to maximize the word-level performance[5]. For ensemble models, we simply averaged the quality scores given by two different models (and then obtained the word-level tags based on the threshold). For the implementation details of the baselines, please refer to Appendix A.3.

4.2 Results

We first compared our self-supervised QE method with different variants of SyntheticQE (Tuan et al., 2021) on the WMT 2019 sentence- and word-level QE tasks. The experimental results are shown in Table 3.

For single models, the baseline SyntheticQE-MT outperforms another baseline SyntheticQE-MLM except on the En-Ru sentence-level task. Our single model consistently outperforms both baselines on both sentence- and word-level tasks in two different language pairs. Additionally, our single model achieves competitive or better performance compared to the highly complex ensemble model

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

[5] For the thresholds used in the experiments, please refer to Appendix A.2.

Method	En-De				En-Ru			
	Sent-Level		Word-Level		Sent-Level		Word-Level	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Results of Supervised Models								
Supervised*	0.473	0.507	0.366	0.396	0.495	0.517	0.410	0.448
Results of Single Unsupervised Models								
SyntheticQE-MT	0.478	0.425	0.349	0.338	0.201	0.233	0.263	0.265
SyntheticQE-MLM	0.386	0.368	0.318	0.309	0.204	0.284	0.181	0.208
Ours	**0.504**	**0.463**	**0.381**	**0.383**	**0.242**	**0.435**	**0.318**	**0.338**
Results of Ensemble Unsupervised Models								
SyntheticQE-MT Ensemble	0.488	0.428	0.360	0.339	0.212	0.246	0.274	0.297
SyntheticQE-MLM Ensemble	0.407	0.379	0.318	0.307	0.210	0.299	0.185	0.216
SyntheticQE-MT+MLM	0.508	0.460	0.373	0.362	0.247	0.317	0.262	0.286
Ours Ensemble	**0.518**	**0.462**	**0.395**	**0.385**	**0.248**	**0.453**	**0.318**	**0.359**

Table 3: Comparison with SyntheticQE (Tuan et al., 2021) on the WMT 2019 sentence- and word-level development and test sets. "*": we followed Kepler et al. (2019) and implemented the supervised models by fine-tuning the multilingual BERT (Devlin et al., 2019). For the implementation details of the supervised models, please refer to Appendix A.4.

Dataset	Method	Sent	Word
SMT	SyntheticQE-MT	0.469	0.417
	SyntheticQE-MLM	0.416	0.298
	Ours	**0.560**	**0.425**
NMT	SyntheticQE-MT	0.526	0.444
	SyntheticQE-MLM	0.424	0.320
	Ours	**0.590**	**0.476**

Table 4: Comparison with SyntheticQE (Tuan et al., 2021) on the WMT 2018 En-Lv test sets.

Method	En-Lv		En-De	En-Ru
	SMT	NMT	NMT	NMT
uMQE	0.385	0.550	0.375	0.243
BERTScore	0.176	0.221	-0.101	0.093
BERTScore++	0.213	0.155	-0.073	0.069
NMT-QE	0.540	0.580	0.452	0.372
Ours	**0.560**	**0.590**	**0.463**	**0.435**

Table 5: Comparison with other previous unsupervised methods (Etchegoyhen et al., 2018; Zhang et al., 2020; Zhou et al., 2020; Fomicheva et al., 2020) on the WMT 2018 En-Lv and the WMT 2019 sentence-level test sets.

SyntheticQE-MT+MLM, which requires both MT and MLM models to generate synthetic data.

For ensemble models, the ensemble model SyntheticQE-MT+MLM outperforms SyntheticQE-MT and SyntheticQE-MLM (including their ensemble variants) in most cases. Our ensemble model performs better than our single model in most cases and consistently outperforms all ensemble baselines.

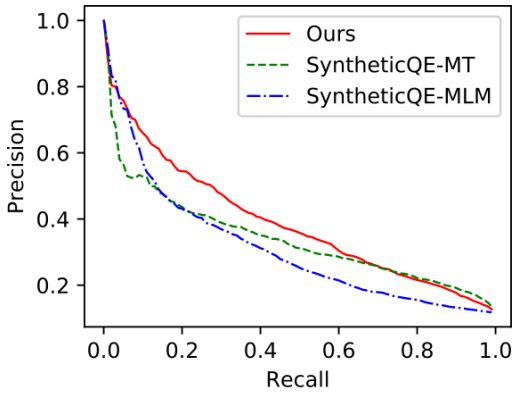


Figure 4: Precision-recall curves of SyntheticQE (Tuan et al., 2021) and our self-supervised method for "BAD" class on the WMT 2019 En-De word-level development set.

To further validate whether our method can generalize across different domains, we conducted experiments on the WMT 2018 En-Lv task, which is in the biomedical domain. As shown in Table 4, our single model outperforms both SyntheticQE-MT and SyntheticQE-MLM on both sentence- and word-level tasks, which confirms that our method can generalize well across different domains.

We also compared our method with other unsupervised sentence-level methods. As shown in Table 5, our method also outperforms other unsupervised methods on sentence-level tasks.

Source	switch between the snapshots to find the settings you like best .
Target & Golden	wechseln Sie zwischen den *Schnappschüsse* , um die gewünschten Einstellungen zu finden .
SyntheticQE-MT	wechseln Sie zwischen den Schnappschüsse , um die *gewünschten* Einstellungen zu *finden* .
SyntheticQE-MLM	wechseln Sie zwischen den Schnappschüsse , um die gewünschten Einstellungen zu finden .
Ours	wechseln Sie zwischen den *Schnappschüsse* , um die gewünschten Einstellungen zu finden .

Table 6: Example of word-level QE using different methods. Erroneous target words annotated in the golden data or detected by the models are highlighted in red and italic.

4.3 Further Comparison with SyntheticQE

To analyze the advantages of our method, we conducted further analysis on the WMT 2019 En-De word-level development set and plot the precision-recall curves for the "BAD" class by setting different thresholds for SyntheticQE and our method. As shown in Figure 4, between the two baseline systems, the precision of SyntheticQE-MT is relatively low when the recall is below 0.2, and the precision of SyntheticQE-MLM is relatively low when the recall is above 0.2. Compared with the baselines, our method reaches a relatively high precision whenever the recall is low or high.

In SyntheticQE-MT, the target side of the synthetic data is produced by MT models, and thus more tokens may be labeled with "BAD" in the synthetic data than in the authentic data since references are less similar to machine-translated sentences than post-editions (Snover et al., 2005). In other words, some "BAD" labels in the synthetic data do not represent erroneous target words but represent words merely different from the expressions in the references. These two types of "BAD" labels cannot be significantly distinguished in the synthetic data, which may be harmful to the model's ability for detecting real errors and finally lead to lower precision when the recall is low.

In SyntheticQE-MLM, the target side of the synthetic data is produced by MLMs, and thus more catastrophic errors appear in synthetic target sentences than in machine-translated sentences (Tuan et al., 2021). In this case, the model mainly focuses on detecting rare catastrophic errors in the target sentences, but is incapable of detecting common subtle errors. Therefore, SyntheticQE-MLM reaches a relatively high precision when the recall is low but a relatively low precision when the recall is high.

ID	WWM	MC Dropout	Sent	Word
1	×	√	0.465	0.344
2	√	×	0.479	0.376
3	√	√	**0.504**	**0.381**

Table 7: Ablation studies on the WMT 2019 En-De development set.

By contrast, our self-supervised QE method does not rely on these noisy synthetic data. Thus our method is not affected by the noise and achieves better results whenever the recall is low or high.

Case study. To further show the advantages of our method, we provide an example in Table 6. In this example, the only erroneous word in the target sentence is "Schnappschüsse", which is corrected to "Schnappschüssen" in the post-edition. SyntheticQE-MT fails to detect this error, and wrongly predicts two correct words "gewünschten" and "finden" as erroneous. SyntheticQE-MLM also fails to detect this subtle error. Our method successfully detects the error while it does not predict other correctly translated words as erroneous.

4.4 Ablation Studies

To compare and analyze the performance of our method with different configurations, we conducted ablation studies on the WMT 2019 En-De development set. The experimental results are shown in Table 7.

Effect of masking strategies. To measure the effect of masking strategies, we conducted experiments using different masking strategies and compared their performances. According to the results, the model with WWM (row 3) outperforms its counterpart without WWM (row 1). Table 8 shows an example of word-level QE using models

Source	in a text box , delete the option text .
Target & Golden	lschen Sie den *ausgewählten Text* in einem Textfeld .
w/o WWM	lschen Sie den ausgewählten Text in einem Textfeld .
w/ WWM	lschen Sie den *ausgewählten* Text in einem Textfeld .

Table 8: Example of word-level QE using different masking strategies. Erroneous target words annotated in the golden data or detected by the models are highlighted in red and italic.

with different masking strategies. In this example, the model without WWM fails to detect the erroneous target word "ausgewählten", which consists of 2 subwords "ausgewählt" and "##en". However, the model with WWM successfully detects this error. This indicates that WWM helps estimate the translation quality of words with multiple subwords.

Effect of MC Dropout. To measure the effect of MC Dropout, we conducted experiments without MC Dropout (row 2) and compared them with their counterparts with MC Dropout (row 3). Experimental results show that the performance decline with the absence of MC Dropout. Additionally, we also try applying MC Dropout to SyntheticQE, but we find no significant improvement over its counterpart without MC Dropout.

5 Related Work

Our work is closely related to two lines of research: (1) quality estimation for machine translation, and (2) masked language models.

5.1 Quality Estimation for Machine Translation

QE aims to evaluate the quality of machine-translated sentences without references, which has been studied mainly under supervised settings. Specia et al. (2013) propose a feature-based QE method using various manually designed features and traditional machine learning models. With the recent prevalence of deep learning, various neural methods for QE have been proposed (Kim et al., 2017; Ive et al., 2018; Fan et al., 2019). Recently, with the development of pretraining, multilingual pretrained language models (Devlin et al., 2019; Conneau and Lample, 2019; Conneau et al., 2020) are also utilized in QE (Kim et al., 2019; Kepler et al., 2019; Moura et al., 2020; Ranasinghe et al., 2020; Rubino and Sumita, 2020; Zhang and van Genabith, 2020; Lee, 2020).

Due to the data scarcity problem in QE, several studies have endeavored to construct unsupervised QE models. For example, Etchegoyhen et al. (2018) build unsupervised QE models using lexical translation tables and language models. Zhang et al. (2020) utilize lexical similarities based on word vectors. Zhou et al. (2020) propose an enhanced version of Zhang et al. (2020), which also utilizes explicit cross-lingual patterns obtained from word alignments and multilingual MLMs. Fomicheva et al. (2020) use different features extracted from NMT models. Tuan et al. (2021) train unsupervised QE models using synthetic data. However, these works are either limited to sentence-level tasks, or negatively affected by the noisy synthetic data. By comparison, our work develops a self-supervised method for both sentence- and word-level QE without using synthetic data.

Our work is also similar to Fan et al. (2019) and Kim et al. (2019). However, their works are designed for supervised QE and require to be fine-tuned on labeled training data, while our work conducts unsupervised QE by directly utilizing the conditional probabilities given by the model and does not require any further fine-tuning process. Moreover, our work utilizes different techniques like WWM (Cui et al., 2019) and MC Dropout (Gal and Ghahramani, 2016) to further improve the performance.

5.2 Masked Language Models

Recently, pretrained masked language models (MLMs) (Devlin et al., 2019) have been widely used in various NLP tasks including natural language understanding (Wang et al., 2019) and machine reading comprehension (Xu et al., 2019). The idea of MLM is also used in other complex NLP tasks. For example, Ghazvininejad et al. (2019) introduce a conditional masked language model (CMLM) for non-autoregressive NMT. Chen et al. (2021) and Zhang and van Genabith (2021) present MLM objectives to improve neural word alignment models. MLM objectives are also used in the training process of supervised QE (Kim et al., 2019; Rubino and Sumita, 2020; Cui et al., 2021). To

the best of our knowledge, our work is the first to utilize MLM objectives for QE under unsupervised settings.

Our work is also similar to translation language modeling (TLM) (Conneau and Lample, 2019). However, TLM is a multilingual pretraining schema designed for fine-tuning on various multilingual downstream tasks, while our work finetunes a multilingual pretrained model on bilingual parallel corpora for unsupervised QE.

6 Conclusion and Future Work

We have presented a self-supervised method for quality estimation of machine-translated sentences. The central idea is to perform quality estimation by recovering masked target words using the surrounding context. Our method is easy to implement and is not affected by noisy synthetic data. Experimental results show that our method outperforms previous unsupervised QE methods. In the future, we plan to extend our self-supervised method to phrase- and document-level tasks.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2018YFB1005103), National Natural Science Foundation of China (No. 62006138, No. 61925601, No. 61772302) and Huawei Noah's Ark Lab. We thank all anonymous reviewers for their valuable comments and suggestions on this work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of International Conference on Learning Representations*.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio L. Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Comput. Linguistics*, 35(1):3–28.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Sheila Castilho, Joss Moorkens, Federico Gaspari, Iacer Calixto, John Tinsley, and Andy Way. 2017. Is neural machine translation the new state of the art? *Prague Bull. Math. Linguistics*, 108:109–120.

Chi Chen, Maosong Sun, and Yang Liu. 2021. Mask-align: Self-supervised neural word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

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*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*.

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 AAAI conference on Artificial Intelligence*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese BERT. *arXiv preprint arXiv: 1906.08101*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Thierry Etchegoyhen, Eva Martínez Garcia, and Andoni Azpeitia. 2018. Supervised and unsupervised minimalist quality estimators: Vicomtech's participation in the wmt 2018 quality estimation task. In *Proceedings of the Third Conference on Machine Translation*.

Kai Fan, Jiayi Wang, Bo Li, Fengming Zhou, Boxing Chen, and Luo Si. 2019. "bilingual expert" can find translation errors. In *Proceedings of AAAI conference on Artificial Intelligence*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33nd International Conference on Machine Learning*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified kneser-ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Chi Hu, Hui Liu, Kai Feng, Chen Xu, Nuo Xu, Zefan Zhou, Shiqin Yan, Yingfeng Luo, Chenglong Wang, Xia Meng, Tong Xiao, and Jingbo Zhu. 2020. The niutrans system for the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*.

Julia Ive, Frédéric Blain, and Lucia Specia. 2018. deepquest: A framework for neural-based quality estimation. In *Proceedings of the 27th International Conference on Computational Linguistics*.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M. Amin Farajian, António V. Lopes, and André F. T. Martins. 2019. Unbabel's participation in the wmt19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation*.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*.

Hyun Kim, Joon-Ho Lim, Hyun-Ki Kim, and Seung-Hoon Na. 2019. Qe bert: Bilingual bert using multi-task learning for neural quality estimation. In *Proceedings of the Fourth Conference on Machine Translation*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

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*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

João Moura, miguel vera, Daan van Stigt, Fabio Kepler, and André F. T. Martins. 2020. Ist-unbabel participation in the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. ESCAPE: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.

Maja Popović. 2012. Morpheme- and pos-based IBM1 and language model scores for translation quality estimation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. Transquest at wmt2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*.

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*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Matthew Snover, Bonnie J. Dorr, R. Schwartz, L. Micciulla, and R. Weischedel. 2005. A study of translation error rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Lucia Specia. 2011. Exploiting objective annotations for minimising translation post-editing effort. In *Proceedings of the 15th Annual conference of the European Association for Machine Translation*.

Lucia Specia, Kashif Shah, José G. C. 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*.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*.

Zhixing Tan, Jiacheng Zhang, Xuancheng Huang, Gang Chen, Shuo Wang, Maosong Sun, Huanbo Luan, and Yang Liu. 2020. THUMT: an open-source toolkit for neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas*.

Jrg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*.

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*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv: 1609.08144*.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jingyi Zhang and Josef van Genabith. 2020. Translation quality estimation by jointly learning to score and rank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Jingyi Zhang and Josef van Genabith. 2021. A bidirectional transformer based alignment model for unsupervised word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *Proceedings of International Conference on Learning Representations*.

Lei Zhou, Liang Ding, and Koichi Takeda. 2020. Zero-shot translation quality estimation with explicit cross-lingual patterns. In *Proceedings of the Fifth Conference on Machine Translation*.

A Appendix

A.1 Detailed Process of Calculating Quality Scores with Monte Carlo Dropout

See Algorithm 1.

A.2 Thresholds Used in Our Method

We used $\tau = 0.385$ for En-De, $\tau = 0.059$ for En-Ru, $\tau = 0.660$ for En-Lv (SMT) and $\tau = 0.616$ for En-Lv (NMT).

A.3 Implementation Details of Baselines

Implementation Details of SyntheticQE

For SyntheticQE-MT, the target side of the synthetic data was produced in a cross-validation setting similar to Negri et al. (2018). The synthetic target sentences were translated using Moses (Koehn et al., 2007) (for SMT datasets) or THUMT (Tan et al., 2020) (for NMT datasets). Specifically, for Moses, we mainly followed the default training process and configurations. We removed sentences longer than 100 words before training. For the language models used in Moses, we used 3-gram Kneser-Ney language models (Heafield et al., 2013). For THUMT, we used the Transformer (Vaswani et al., 2017) architecture with base setting for NMT models. We used the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$ to optimize model parameters. We used the same learning rate schedule as Vaswani et al. (2017) with 4,000 warmup steps. During training, we set the batch size to 25,000 tokens, the number of training steps to 100,000, the penalty of label smoothing to 0.1 and the dropout rate to 0.1. We performed subword segmentation using BPE (Sennrich et al., 2016) with 32,000 merge operations.

For SyntheticQE-MLM, we followed Tuan et al. (2021) and produced the target side of the synthetic data by randomly substituting, deleting, and inserting words. The substitutions and insertions were performed using MLMs. Since our experiments were conducted on datasets in different domains, the MLMs we used were obtained by fine-tuning the multilingual BERT (Devlin et al., 2019) on the target side of the parallel corpora.

The TER (Snover et al., 2005) scores of the synthetic training data and the authentic development and test data are shown in Table 9.

For the QE models in SyntheticQE, we followed Kepler et al. (2019) and used a BERT-based model for both sentence- and word-level tasks. The models were fine-tuned on the synthetic data. For the optimizer, we used the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. We set the batch size to 128, the maximum sequence length to 256, the number of training steps to 100,000, the learning rate to 5×10^{-5} and the dropout rate to 0.1. We evaluated our model every 1,000 steps and chose the model with the best performance on the development set for inference. We tuned the threshold on the development set to maximize word-level performance.

Implementation Details of Other Unsupervised Sentence-Level Baselines

For uMQE (Etchegoyhen et al., 2018), we set the minimal prefix length to 4, the maximal number of candidates in the translation table to 4 and the order of the language model to 5.

For the word embeddings in BERTScore (Zhang et al., 2020) and BERTScore++ (Zhou et al., 2020), we used the contexutalized embeddings in the 9th layer of the multilingual BERT (Devlin et al., 2019). In BERTScore++, we set a to 0.8 and λ to 0.01.

For NMT-QE (Fomicheva et al., 2020), we use the D-TP measure for unsupervised QE. This measure uses MC Dropout (Gal and Ghahramani, 2016) to calculate the expectation of sentence-level translation probabilities. For the NMT models used in NMT-QE, we implemented it based on THUMT (Tan et al., 2020) with the base setting as presented in Vaswani et al. (2017). We evaluated our model every 1,000 steps and chose the model with the best performance on the development set for inference. For the MC Dropout process, we set $N = 30$.

A.4 Implementation Details of Supervised Models

The supervised models were also implemented based on the multilingual BERT (Devlin et al., 2019). We used the official training data provided by WMT to train the models. Each model were trained for 5 epochs. We set the batch size to 12 and the learning rate to 10^{-5}. We tuned the threshold on the development set to maximize word-level performance.

Algorithm 1 Calculating quality scores with Monte Carlo Dropout

Input: source sentence $\mathbf{x}$, target sentence $\hat{\mathbf{y}} = (\hat{y}_1, \cdots, \hat{y}_T)$, number of samples for each target token N, number of estimations N', model parameter $\boldsymbol{\theta}$

Output: quality scores of all target tokens $score(\hat{y}_1), \cdots, score(\hat{y}_T)$

for $n \leftarrow 1$ to N' **do**
    $\hat{\mathbf{y}}_m^{(n)} \leftarrow \emptyset$
for $t \leftarrow 1$ to T **do**
    $score(\hat{y}_t) \leftarrow 0$
    Randomly sample N integers $n_1, n_2, \cdots, n_N$ from $[1, N']$
    for $i \leftarrow 1$ to N **do**
        $\hat{\mathbf{y}}_m^{(n_i)} \leftarrow \hat{\mathbf{y}}_m^{(n_i)} \cup \{\hat{y}_t\}$
for $n \leftarrow 1$ to N' **do**
    $\hat{\mathbf{y}}_o^{(n)} \leftarrow \hat{\mathbf{y}} \backslash \hat{\mathbf{y}}_m^{(n)}$
    Sample a model $\hat{\boldsymbol{\theta}}_n$ from $\boldsymbol{\theta}$ using dropout
    Calculate $P(\hat{y}_t|\mathbf{x}, \hat{\mathbf{y}}_o^{(n)}; \hat{\boldsymbol{\theta}}_n)$ for all $\hat{y}_t \in \hat{\mathbf{y}}_m^{(n)}$ using the model $\hat{\boldsymbol{\theta}}_n$
    for each $\hat{y}_t \in \hat{\mathbf{y}}_m^{(n)}$ **do**
        $score(\hat{y}_t) \leftarrow score(\hat{y}_t) + P(\hat{y}_t|\mathbf{x}, \hat{\mathbf{y}}_o^{(n)}; \hat{\boldsymbol{\theta}}_n) \ / \ N$
return $score(\hat{y}_1), \cdots, score(\hat{y}_T)$

Dataset	**En-Lv**		**En-De**	**En-Ru**
	SMT	NMT	NMT	NMT
Train (SyntheticQE-MT)	0.452	0.418	0.453	0.662
Train (SyntheticQE-MLM)	0.292	0.292	0.319	0.387
Dev**	0.200	0.280	0.141	0.127
Test**	0.200	0.300	0.168	0.154

Table 9: TER scores of the synthetic training data and the authentic development and test data. "**": the TER scores are computed using the human post-editions instead of the references.