Word-Level Quality Estimation for Korean-English Neural Machine Translation

SUGYEONG EO¹, CHANJUN PARK¹,², HYEONSEOK MOON¹, JAEHYUNG SEO¹, AND HEUISEOK LIM¹

¹Department of Computer Science and Engineering, Korea University, Seoul 02841, Republic of Korea
²Upstage, Yongin-si, Gyeonggi-do 16942, Republic of Korea

Corresponding author: Heuiseok Lim (limhseok@korea.ac.kr)

This work was supported in part by the Institute of Information Communications Technology Planning Evaluation (IITP) funded by the Korean Government through the Ministry of Science and ICT (MSIT), South Korea, through the Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques under Grant 2020-0-00368; and in part by MSIT under the Information Technology Research Center (ITRC) Support Program Supervised by the Institute for Information Communications Technology Planning Evaluation (IITP) under Grant IITP-2018-0-01405.

ABSTRACT Quality estimation (QE) task aims to predict the machine translation (MT) quality well by referring to the source sentence and its MT output. The various applicability of QE proves the importance of QE research, but the enormous human labor to construct the QE dataset remains a challenge. This study proposes three automatic word-level pseudo-QE data construction strategies using a monolingual or parallel corpus and an external machine translator without human labor. We utilize these individual pseudo-QE datasets to finetune multilingual pretrained language models such as cross-lingual language models (XLM), XLM-RoBERTa, and multilingual BART and comparatively analyze the results. Considering the synthetic dataset creation setup, we attempt to validate the objectivity of the QE model by leveraging four test sets translated by external translators from Google, Amazon, Microsoft, and Systran. As a result, XLM-R-large shows the best performance among mPLMs. We also verify the reliability of the QE model through the close performance gaps between different test sets. To the best of our knowledge, this is the first study to experiment with word-level Korean-English QE.

INDEX TERMS Quality estimation, neural machine translation, multilingual pretrained language model, natural language processing.

I. INTRODUCTION

Owing to advances in neural machine translation (NMT), interest in measuring the quality of machine translation (MT) output is growing significantly. Quality estimation (QE) is the task of predicting quality of MT output by accessing the source sentence and MT output without a reference translation. Unlike the existing MT evaluation metrics, accurate quality measurement is not guaranteed in QE. However, QE study is continuously being conducted due to the characteristic that it not only reduces the burden of human labor by not requiring reference translation, but also has many applications in the real world. In accordance with the need for such research, the Conference on Machine Translation (WMT), an authorized workshop in the field of MT, has been holding QE tasks annually since 2012.

Another distinguishing feature of QE is that it can be used in various ways for each level of granularity, such as document, sentence, and word. First, at the document level, the span indicates incorrect part of the document with respect to the MT output. In addition, the level of translation errors are identified through the classification on quality annotations for translation errors into minor/major/critical.

At the sentence level, the degree of translation error is evaluated as human translation error rate [33] or direct assessment (DA) score in numerical value. These values are used to measure the reliability of the MT output, to decide whether the additional post-editing is needed, and to filter out low quality results. Moreover, users can also rank the MT outputs from multiple machine translators according to their translation quality [35].

At the word level, MT output is tagged as OK or BAD for each token, of which only the tokens attached as BAD can be post-edited to avoid over-correction [21]. Furthermore,
quality annotations are marked to the tokens of the aligned source sentence. Hence, QE system users who are not fluent to the target language can easily identify incorrected results.

Contrary to the advantages of QE, there exists a paradox in terms of QE dataset construction [6]. In the process of building QE datasets, post-edited sentences that are edited by translation experts are required. For generating this post-edited sentences, the translation experts must identify the error part of the translation result and change it to the correct sentence through minimal correction. Hence, more effort may be even required than the cost of building parallel corpus.

To alleviate this QE dataset construction issue, we propose three efficient strategies to construct word-level pseudo-QE dataset with only a corpus and an external machine translator. Unlike existing manual QE dataset production method, these are the effects of our dataset construction strategies: First, no human intervention is required in terms of data construction. Second, considering language resources, mono and parallel corpora are utilized respectively. Third, even in low-resource language setting where it is difficult to hire a translation expert, QE data generation is possible even if corresponding corpus and Google translator are supported. In our experiment, we conduct a word-level Korean-English QE study in a low-resource setting.

To compare the performance of pretrained language models on our three word-level QE dataset construction strategies, we finetune multilingual pretrained language models (mPLMs) that learned Korean and English together, such as cross-lingual language model (XLM) [20], XLM-RoBERTa (XLM-R) [3], and multilingual BART (mBART) [24]. When validating our QE models, we test QE models using the MT results provided by Google, Amazon, Microsoft, and Systran to demonstrate that the QE model objectively predicts the translation quality properly. The following are the contributions of this study:

- To handle the limitations of QE in the data construction, we propose three pseudo-QE dataset construction strategies for word-level Korean-English QE learning utilizing only existing dataset and an external translator without additional costs.
- With our experiments and comparative analysis on various mPLMs with the constructed QE dataset, we introduce the model with excellent performance in Korean-English QE.
- We confirm that the QE model objectively predicts the quality of MT output after validity test using various external translators.
- To the best of our knowledge, while English-Korean QE has been attempted before [14], [16], this study is the first to conduct word-level Korean-English QE.

II. RELATED WORKS AND BACKGROUND

A. QUALITY ESTIMATION (QE)

Most traditional QE studies have focused mainly on feature extraction or selection. In general cases, machine learning algorithms such as Gaussian process, support vector machine, and regression trees are adopted for the feature selection [1], [34]. Additionally, pseudo-reference or linguistic features are extracted through external resources, such as a linguistic parser, tagger or named entity recognizer, for the feature extraction [9], [25]. However, these studies aim to finding the complex relationships between the corresponding features and reference features for QE. Thus, utilization of such processes leads to several limitations as the heuristic process in selecting optimized features is required.

With the advent of deep learning, several studies on neural QE are introduced. recurrent neural network [2] and long short-term memory [10] are mostly utilized in WMT16 [13], [29]. In WMT17, [15] propose a novel predictor-estimator structure consisting of word prediction and QE learning model, and obtain state-of-the-art (SOTA) performance in all subtasks of QE.

Owing to the paradigm shift toward using pretrained language models to understand large-scale mono-text [4], most QE studies rely on the mPLMs. In WMT19, Unbabel [12] adopts predictor-estimator structure and replaces the predictor model in the pretrained BERT [4] or XLM [20] structure; ETRI also constructs a QE model based on mBERT [17].

In most studies on WMT20 exploit mPLMs, which is composed of pretrained and finetuening phase. TransQuest proposes two model structures, MonoTransQuest and SiameseTransQuest, for sentence-level direct assessment of Sub-task 1 and sentence-level post-editing effort of Sub-task 2; the former achieves the highest score [30]. In the study, finetuning is carried out on the pretrained XLM-R structure in MonoTransQuest [3]. In SiameseTransQuest, source sentence and target sentence are fed into two separate XLM-R models and finetuned. Furthermore, TransQuest suggests word-level QE model MicroTransQuest by adopting XLM-R and finetuning the corresponding model to predict the OK/BAD tag for each token in the MT output [31].

Similar to TransQuest, Bering lab introduces a two-stage training phase that utilizes XLM-R as a pretrained model and obtains comparable performance in English-German sentence- and word-level post-editing effort of Sub-task 2 [22]. The model pretrain with an augmented data using parallel corpus, then finetune with WMT official dataset.

Huawei Translation Service Center shows SOTA performance in English-German sentence level QE as well as English-Chinese word-level QE without adopting any of the mPLMs [39]. It suggests a transformer-based NMT model as a predictor and a task-specific regressor or classifier as an estimator. In the training process, a bottleneck adapter layer [11] is newly added to improve the efficiency of transfer learning and to prevent from overfitting.

In summary, the trend of QE research has largely moved from the approach of rule and statistic models that extract or select optimized features [1], [25] and deep learning-based approach [13], [29] to focusing on the utilization of mPLMs [17], [22], [26]. However, these studies mainly focus on specific language pairs, with little emphasis on other
language pairs. Multilingual QE [31], [37] and zero-shot QE [31], [41] are being widely researched. However, the development of the QE model that focuses on Korean-English rarely exists. Although [16] proposed a predictor-estimator structure and constructed an English-Korean QE dataset, a word-level QE dataset or model has not been proposed for Korean-English.

B. WORD-LEVEL QE
A word-level QE task aims to indicate incorrectly translated tokens in the MT output as well as corresponding source sentence. Mistranslated tokens are expressed by the OK/BAD tag for efficient post-editing. To train word-level QE model, a task-specific dataset including a source sentence, MT output, and their respective word-level quality annotations is required. Typically, word-level quality annotations are marked based on the edit distance, quantifying the number of errors such as insertions, substitutions, and deletions between the MT output with a human post-edited sentence [36]. Additionally, considering the missing words in a sentence, the gap tag is inserted before and after each token of the MT output. The corresponding position is annotated as BAD tag if there are any missing words. Annotations in the source sentence token are attached based on alignment with the MT output, and the gap tag is not appended in this case.

The word-level QE model is trained to predict the annotation of each token in a sentence by referring to the source sentence and MT output. The general training process of QE consists of two steps. We generate QE data in the first step, and then we perform QE modeling based on the constructed data.

III. PROPOSED KOREAN-ENGLISH WORD-LEVEL QE MODEL
Our proposed Korean-English word-level QE method consists of two steps. We generate QE data in the first step, and then we perform QE modeling based on the constructed data.

A. STEP 1: PSEUDO-QE DATA GENERATION
1) MOTIVATION
To build word-level QE data, post-edited sentences are required. However, it is quite limited to build large-scale data due to the human labor and time cost. Even in a low-resource setting, building a QE dataset is more challenging because of a limitation in obtaining the parallel corpus itself and recruiting a translation expert. More importantly, even if the dataset built with the post-edited sentence is used in QE and automatic post editing field, much extensions to another task is hardly expected compare to the parallel corpus. Hence, there is a limitation in generating large amounts of data.

With the motivation, we introduce three automatic dataset construction strategies in generating word-level Korean-English pseudo-QE datasets. To meet the data requirements, monolingual corpus in target language or parallel corpus and an external machine translator are leveraged. The following are strategies of three pseudo-dataset construction:

- **M-based** strategy: Leveraging fully monolingual corpus.
- **P-based** strategy: Leveraging fully parallel corpus.
- **H-based** strategy: Hybrid utilization of monolingual and parallel corpus.

In the M-based and P-based methods, monolingual and parallel corpora are utilized for dataset production, respectively, whereas in the H-based method, both corpora are partially utilized to build QE data. All strategies are translated by an external machine translator. Therefore, we use Google translator, which is easily accessible and has a large number of users.

2) DATA GENERATION WITH M-based STRATEGY
In the process of constructing the QE dataset based on the M-based strategy, we first translate a monolingual corpus of the target language (English) into the source language (Korean).

In the second phase, the backward translated Korean sentence is converted into English data again through a forward translation process. We intentionally utilize round-trip translated English sentence with errors as MT output since errors usually exist in most of the translation. These errors are differ from those of appeared in the process of translating the right source sentence into forward translation. In the third phase, we regard the English monolingual corpus as a post-edited sentence, and then measure the Levenshtein algorithm-based edit distance comparing the MT output and post-edited sentence.

Finally, BAD tags are attached to tokens that need to be corrected based on the measured edit distance, and OK tags are attached to correct tokens. Gap tags are also based on edit distance. If some are missing, BAD annotation is attached to the location of the gap token. Annotating the source token is based on alignment. Using the Korean-English alignment model, word alignment is performed between the pseudo-source sentence backward translated and its MT output from forward translations. Thereafter, the BAD tag is annotated to the source token aligned with the MT output token to which the BAD tag is attached.

3) DATA GENERATION WITH P-based STRATEGY
When constructing a QE dataset with parallel corpus, the source sentence (Korean) is first translated forward into the target language (English). The translated sentence has an error committed, which is used as an MT output among QE dataset requirements.

In the second process, we measure the edit distance for the MT output and the target-side of the parallel corpus, which is considered as a post-edited sentence. The error token is tagged based on the edit distance, and in the case of missing words, the BAD tag is attached to the place corresponding to the gap token in the third process. In the case of source token tags, we use the alignment model. If there is a BAD tag among
the MT output tokens, we also tag the aligned source tokens with BAD. We mark all remaining token positions without BAD tags with OK tags.

4) DATA GENERATION WITH H-based STRATEGY
In the M-based strategy, because both the pseudo-source sentence and its MT output are generated by translation systems, errors exist inevitably in the two sentences. To handle this problem, we intend to observe how the result will be different when the source side is replaced with the correct sentence. Equivalently, we propose an H-based approach that utilizes data from both monolingual and parallel corpora.

With the H-based strategy, we compose the source-side text of the parallel corpus as the source sentence. In addition, we configure the target language text after a round trip as MT output, as in the M-based strategy. For the H-based strategy, the source text and the MT output are not directly translated results. In this regard, there is a possibility that the semantic connection between the MT output and the source sentence may be weakened. In the case of tag attachment, it proceeds in the same manner as other strategies. After measuring the edit distance between the MT output and the monolingual corpus, we tag the MT annotation, align the source sentence and the MT output, and annotate source tags.

B. STEP 2: WORD-LEVEL QE MODELING
1) PRETRAINED MODELS
We train a model on a pseudo-QE dataset with MicroTransQuest [31], which performs word-level QE among TransQuest’s open-source framework [30]. The model is based on XLM-R in the finetuning process. We additionally leverage mBART [24] and XLM-MLM, both of which support Korean and English.

a: CROSS-LINGUAL LANGUAGE MODEL
In XLM [20], language understanding is trained in a self-supervised learning manner. Equivalently, by utilizing intact monolingual or parallel corpus, noise is intentionally injected, and the model learns to denoise it as the original corpus. There are three main objectives used in XLM: causal language modeling (CLM), masked language modeling (MLM), and translation language modeling (TLM), each of which is used to learn about language.

The first of the three training objectives, CLM, aims to predict the following words (x_t) by referring to the previous words (x_1, x_2, ..., x_t-1), like P(x_t | x_1, x_2, ..., x_t-1; θ).

MLM is based on the Cloze task [38], which is a pretrained noising scheme from [4]. Some tokens corresponding to 15% of sentence pairs are randomly replaced with the [MASK] token, and the model is trained to predict these tokens as original tokens. By referring to both the contexts before and after the masked token, the token of the corresponding position is predicted. This feature compensates for the disadvantages of the preceding CLM. Here, 80% of the 15% masking probability is replaced with the masking token, 10% is replaced with another token, and the remaining 10% remained unchanged. One of the two things that XLM distinguishes from the MLM objectives of [4] is that it utilizes a stream of text instead of pairs of sentences. The unit of input is 256 tokens, not sentence pairs, and it is split and composed of individual inputs when it reaches the max token range. In addition, another difference of XLM is that tokens are sampled according to a multinomial distribution in the text stream. In this case, the sampling weight is equal to the square root of the inverted frequencies.

In contrast to other objectives that utilize monolingual corpus, parallel corpus is needed for TLM. The TLM objective is an extended version of MLM as a pair of language and is proposed to improve cross-lingual understanding. Random masking is performed for each side in the concatenated source and target sentence, and then they are predicted into the original sentence. In this process, the masked token of the source side refers not only to surrounded tokens but also to the content of target side. Thus, the model is induced to learn alignment information for language pairs. In particular, in the QE task, which requires understanding between languages, the TLM objective have a positive effect on performance improvement [7].

When training the QE model, we use only the MLM model learned with Wikipedia corpus for a total of 100 languages, including Korean and English. Because models with other objectives do not learn Korean together, we do not apply these for QE training.

b: XLM-RoBERTa
XLM-R [3] is an extension of XLM learning with only MLM objective out of the three strategies of XLM, as a pretraining process. In contrast to Wikipedia, CommonCrawl corpus leveraged in XLM-R contains a much larger amount, especially in low-resource languages. However, when the actual amount of data is significantly expanded, changes in vocabulary and sampling size in fixed model capacity setting cause a performance trade-off between high-resource language and low-resource language. To address this limitation, the number of parameters in the model, that is, the model capacity, is expanded. With this feature, XLM-R shows superior performance for low-resource languages.

When training the model, batch sampling is performed for each language to reduce bias in high-resource languages. For i-th language, sentences are sampled according to a multinomial distribution with probability $\lambda_i$ as in Equation 1

$$\lambda_i = \frac{\delta_i}{\sum_{j=1}^{n} \delta_j}, \quad \delta_i = \frac{l_i}{\sum_{k=1}^{n} l_k}$$

In contrast, [20] set $\alpha = 0.5$ when sampling a language, while XLM-R set it to 0.3. Model training proceeds in the
same manner as XLM. However, language embeddings are not used in training to facilitate code-switching.

c: MULTILINGUAL BART

mBART [24] is a multilingual extension of BART [23], pre-trained in a total of 25 languages. In the pretraining process, mBART adopted text infilling and sentence permutation among BART’s noising schemes to perform self-supervised learning. The former replaces sequence of tokens with one masking token; hence, MLM is extended to multiple tokens. In the substitution process, the span length is determined based on Poisson distribution, and the model predicts masked tokens in consideration of the bidirectional context. In this process, the model not only understands the language but also recognizes the number of masked tokens. In the latter case, sentence permutation refers to randomly changing the order of sentences. In this process, the model understands the language between sentences.

mBART is learned to maximize the log likelihood

\[ L_0 = \sum_{c \in C} \sum_{i \in C} \log P(t|H(t); \theta) \]

so that the text \( t \) of the document \( C \) composed of \( i \)-th language is best restored to the original state for the problematic sentence caused by the noising function \( H \). In the learning process, the number of resources for each language is balanced by performing up/down sampling in consideration of low-resource languages.

2) FINETUNING FOR QE

Once word-level QE data are built, we train a QE model based on these data. For the input, first, we add gap token \( g \) before and after MT output \( T_j = \{t_1, t_2, \ldots, t_q\} \) to compose input \( G_j = \{g, t_1, g, t_2, \ldots, g, t_q, g\} \). Therefore, the number of tokens as a result of adding \( g \) tokens is \( |G_j| = 2|T_j| + 1 \). Thereafter, we concatenate the source sentences \( S_j = \{s_1, s_2, \ldots, s_p\} \) and \( G_j \) into [SEP] tokens and add a [CLS] token to the front. For each statement in the QE dataset \( D = \{S_j, T_j, \tilde{S_j}, \tilde{G_j}\} \), we construct the input structure as input in Equation 2.

\[
\text{input} = [\text{CLS}], s_1, s_2, \ldots, s_p, [\text{SEP}], g, t_1, g, t_2, g, \ldots, g, t_q, g
\]  

(2)

Here, \( |S_j| = |\tilde{S_j}|, |G_j| = |\tilde{G_j}| \) is established as the correct answer tag \( \tilde{S_j} \) corresponding to the source token and for the correct answer tag \( \tilde{G_j} \) as a result of the MT output with gap token. The result corresponding to the location of each token passes through the softmax layer, and the cross-entropy loss for each token is minimized.

We depicted the entire process of data construction and QE model learning in Figure 1. The method of synthesizing the data required for QE learning using monolingual and parallel corpus is shown in the lower left and the structure of the corpus obtained as a result of each process for each strategy is in the upper center. Thereafter, each model is changed to an input structure suitable for training and the learning proceeds.

IV. EXPERIMENTS AND RESULTS

A. DATASET DETAILS

We used the parallel corpus released by AI HUB.\(^1\) For a fair comparison between our proposed methods, we extracted target sentences from the corresponding corpus and regarded them as a monolingual corpus. The amount of training and validation dataset is 96K and 12K, respectively, and we generated pseudo-QE data from them. In constructing the QE dataset, we adopted word-level corpus builder from Unbabel,\(^2\) and utilized Tercom software version 0.7.25 [33] for the edit distance computation. Separation of special characters (e.g., punctuation) and case sensitivity was done by the Mosesdecoder [18]. After training word-level alignment using the parallel corpus on AI HUB using FastAlign [5], we proceeded with alignment of the QE data.

Test dataset is composed of 12K sentences without overlapping with the training and validation set. We adopt external machine translator including Google,\(^3\) Amazon,\(^4\) Microsoft,\(^5\) and Systran\(^6\) for the translation of source sentences. Performance assessment of the constructed word-level QE model is proceeded following these translation results, and we have attached reference annotation of the test set in the same process in the section III-A3.

B. MODEL DETAILS

For the finetuning process of each mPLM in our study, we exploit TransQuest\(^7\) framework and leveraged XLM,\(^8\) XLM-R,\(^9\) and mBART\(^10\) model structure released by HuggingFace [40]. We used the tokenization and token classification models provided by the corresponding mPLM to segment sentences and train the QE model. Because no suitable token classification model for mBART has been released, we made an appropriate token classification class. Table 1 is the details about mPLMs:

For training the QE model, we measured the cross-entropy loss of each token. Additionally, in evaluation process, we used Matthew correlation coefficient (MCC) and F1-score.

C. MAIN RESULTS

Table 2 shows the results of finetuning with mPLMs on pseudo-QE data sets generated using M-, H- and P-based strategies, and then evaluated with a test set consisting of four MT outputs. We mark the highest performance with an underline when analyzed by the method of data construction and in bold when compared by mPLMs.

1 https://aihub.or.kr/
2 https://github.com/Unbabel/word-level-qa-corpus-builder.git
3 https://translate.google.co.kr/?hl=ko
4 https://aws.amazon.com/translate/
5 https://translate.systran.net/
6 https://translate.microsoft.com/ko-kr/translator/
7 https://github.com/TharinduDR/TransQuest
8 xlm-mlm-100-1280
9 xlm-roberta-base/large
10 facebook/mbart-large-cc25/50
FIGURE 1. Overall word-level pseudo-QE dataset building and learning process.

TABLE 1. Pretraining setting for each mPLMs. From the left, each column indicates the number of languages learned, the type of dataset used, the subword tokenization training strategy, and the number of vocabulary. The total number of parameters of XLM-R-base and XLM-R-large is 270K and 550K, respectively. We denote CommonCrawl corpus as CC, SentencePiece [19] as SPM, and byte pair encoding [32] as BPE.

| mPLMs            | # of langs | Dataset   | Tok      | Voca   |
|------------------|------------|-----------|----------|--------|
| XLM-R (base)     | 100        | CC        | SPM      | 250K   |
| XLM-R (large)    | 100        | CC        | SPM      | 250K   |
| mBART            | 25         | CC        | SPM      | 250K   |
| mBART50          | 50         | CC        | SPM      | 250K   |
| XLM-MLM-100      | 100        | Wikipedia | BPE      | 200K   |

By comparing the performance differences for the four external MT outputs by mPLMs, most of them show almost similar achievements for each QE dataset generation strategy, except for the translation results of Systran. First, in the case of XLM-R-base with M-based strategies, Matthew’s correlation coefficient for MT output (target MCC) ranged from a maximum of 0.382 for Amazon to a minimum of 0.251 for Systran, showing a difference of 0.131. The second-lowest performance is Google's 0.338, narrowing the gap between external machine translators. We also observed that the performance differences of target MCCs in XLM-R-large, mBART, and XLM had a slight margin except for Systran.

From these results, despite that we used Google translator in building the synthetic QE data, we demonstrated that no performance bias exists toward one external translator in the above results.

1) ANALYSIS OF PERFORMANCE ACCORDING TO THE PSEUDO-QE DATASET CONSTRUCTION METHOD

The results of QE finetuning based on the pseudo-QE data construction strategies are as follows. On the Google test set based on the target MCC, the QE performance of the XLM-R-large is 0.364 for M-based, 0.340 for H-based, and 0.401 for P-based. From this result, we confirmed that the QE data using the P-based strategy performed best. Similarly, in Amazon, Microsoft, and Systran, the M-based strategy is 0.417, 0.408, 0.266, the H-based is 0.397, 0.387, 0.226, and the P-based is 0.443, 0.433, 0.306, respectively. The performance of the P-based strategy remains the best among three strategies.

From the results, we observed that the M-based strategy was consistently higher than H-based strategy. For the M-based strategy, we first predicted a performance decline because errors exist to the sentence when translating to the source language. However, the overall experimental results including previous results, show that the M-based strategy is up to 0.135 (mBART50) higher for the target MCC than the H-based one. Since we directly translate pseudo-source sentences into MT output in M-based but not H-based strategy, we expect the weakened semantic connectivity to cause performance degradation.

We also observed that although there is some trade-off between performance and data construction cost, the burden

11The pseudo-QE datasets based on the M-based and the H-based strategy are the same on the target side but not on the source side.
TABLE 2. Results of testing fine-tuned mPLMs using the Pseudo-QE dataset with multiple external translators.

| Translator Method | Google | Amazon | Microsoft | System |
|-------------------|--------|--------|-----------|--------|
|                   | XLM-B | XLM-L | XLM-B | XLM-L | XLM-B | XLM-L | XLM-B | XLM-L | XLM-B | XLM-L | XLM-B | XLM-L | XLM-B | XLM-L | XLM-B | XLM-L |
| Target MCC        | 0.358 | 0.564 | 0.328 | 0.343 | 0.294 | 0.382 | 0.417 | 0.375 | 0.388 | 0.323 | 0.372 | 0.408 | 0.365 | 0.379 | 0.313 | 0.251 | 0.266 | 0.222 | 0.248 | 0.200 |
| Target F1-BAD     | 0.315 | 0.540 | 0.309 | 0.318 | 0.297 | 0.351 | 0.397 | 0.344 | 0.256 | 0.320 | 0.341 | 0.387 | 0.338 | 0.243 | 0.314 | 0.204 | 0.226 | 0.192 | 0.118 | 0.201 |
| Target F1-OK      | 0.379 | 0.401 | 0.377 | 0.341 | 0.326 | 0.411 | 0.443 | 0.403 | 0.409 | 0.348 | 0.404 | 0.433 | 0.403 | 0.403 | 0.333 | 0.291 | 0.304 | 0.287 | 0.285 | 0.239 |

of constructing a parallel corpus can be reduced by leveraging only the M-based strategy. However, if only the entirely monolingual corpus is used to build the QE training data, it can be built at a much larger capacity than the parallel corpus. In addition, the difference in amount is more pronounced when compared to the parallel corpora because M-based strategy requires an English monolingual corpus, not Korean. In this paper, we compared the performance according to each strategy and use the same amount. However, after significantly expanding the amount of monolingual corpus, we expect additional performance improvement of the QE model.

In the case of gap MCC, the overall MCC is relatively low because the gap tag is composed of OK tags in most examples. It reveals almost similar performance for the three approaches, and the P-based strategy shows the lowest performance almost consistently. In the case of P-based strategy, the f1 score for the OK tag is consistently the highest. However, the f1 score for the BAD tag is not, and so we analyze that the training data is slightly overfitted compared with other approaches. In summary, when viewed based on MCC, the result indicates that the P-based approach has a more positive effect on training compared with other strategies.

2) ANALYSIS OF PERFORMANCE ACCORDING TO THE PRETRAINING MODEL

We analyze the performance difference according to the pre-trained model for each mPLM. The model with consistent
superior performance in all target MCC, source MCC, and gap MCC is the XLM-R-large, whereas the model with the worst performance in most of the three indicators is XLM. We rank the performance of mPLMs for each strategy and averaged the rankings. Hence, we confirm that XLM-R-base, mBART, and mBART50 are not ranked by apparent differences. However, there are distinct performance differences in XLM-R-large and XLM. We conducted an analysis of this result from three perspectives: data used in pretraining, tokenization methods, and noising schemes.

**a: DATASET USED FOR PRETRAINING**
First, we scrutinize the performance difference by model from the perspective of the data used during pretraining by mPLMs. [3] noted that CommonCrawl corpus contains much richer data in low-resource languages (including Korean) than Wikipedia and pretraining of mPLM using CommonCrawl corpus performed better in low-resource languages. Consistently, the XLM-R and mBART models pretrained with CommonCrawl corpus exhibit higher finetuning performance than XLM. Therefore, we infer that the data used in the training process of mPLMs influence the QE performance at the word level.

**b: ANALYSIS ACCORDING TO THE TOKENIZATION METHOD**
From the analysis of each model from a tokenization perspective, when sentence segmentation is performed using XLM Tokenizer, a sign indicating the end of the word is added to the end of all syllables, and all onset, nucleus, and coda are separated. For example, given a ‘보요’ [boy band]’, in contrast to other mPLMs that tokenize it into ‘니요’ ‘ 우리나라’ based on the SentencePiece model, XLM’s BPE-based tokenizer tokenizes it into ‘니요’, ‘는 (w)’ ‘, ‘, ‘, ‘, ‘, ‘, ‘, ‘, ‘. Consistent with the results of [27], [28], which showed lower performance when completely separated into Jamo units compared to subword segments, XLM showed the lowest performance. Therefore, the tokenization method is also regarded as a factor influencing performance in word-level QE.

**c: ANALYSIS ACCORDING TO NOISING SCHEME**
Comparing pretraining approaches, additional noising schemes are used in mBART and mBART50, such as text infilling and sentence permutation in contrast to other models that use only MLM for training. From the experiment, we found that mBART performs both generation and classification problems well. However, we infer that the noising scheme of mBART does not play a significant role in improving performance in word-level QE.

3) ANALYSIS THROUGH VISUALIZATION
We examine more precisely how adequately the model predicts the OK/BAD tags. Based on the MT output, we measure true positive, true negative, false positive, and false negative for each data generation strategy and mPLM, and then we visualize these as a heatmap in Figure 2. The color span between 50K and 120K refers to an increase in the amount of data from navy to yellow.

![Figure 2. Results showing true negative (upper left), false negative (bottom left), true positive (bottom right), false positive (upper right) as a heatmap.](image-url)
negative values most distinctly compared to other strategies. The color of the M-based is closer to navy than the P-based strategy. However, it gains more yellow than the H-based in the true negative. In the case of the H-based, most tokens tend to be predicted positively. The outcome of comparing M- and H-based approaches are the basis for the previous description that if only the amount of monolingual corpus is expanded, it could show sufficiently comparable performance. Additionally, consistent with section IV-C2, mPLMs show that the true negative of XLM is significantly lower in the M-based and that the OK/BAD tag is not correctly attached.

V. CONCLUSION
In QE, the quality of the MT output is predicted without the reference translation, and its diverse applicability highlights the importance of QE. In this study, we automatically built a word-level Korean-English pseudo-QE dataset without intervention of a translation expert. Through comparison and analysis of the QE finetuning results of various mPLMs, we confirmed that XLM-R-large has the best performance among mPLMs. Because the dataset is constructed in a synthetic manner, its reliability must be demonstrated in order to use it. Therefore, we validated the objectivity of our finetuned QE model by identifying insignificant performance gaps among multiple external machine translators. This word-level QE dataset construction process requires only monolingual or parallel corpus and external machine translator. For that reason, this process is applicable to any language pair supported by Google translator, with a corresponding corpus for each data building strategy. This study slightly enlarges the dataset size to 10K compared to the WMT QE dataset of 7K, but the size is still limited. Therefore, we plan to scale the amount of QE data in future work.

ACKNOWLEDGMENT
An earlier version of this paper was presented in part at the Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021) and in part at the Annual Conference on Human and Language Technology. (Sugyeong Eo and Chunjun Park contributed equally to this work).

REFERENCES
[1] E. Biçici and A. Way, Referential Translation Machines for Predicting Translation Quality, Stroudsburg, PA, USA: Association for Computational Linguistics, 2014.
[2] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” 2014, arXiv:1406.1078.
[3] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, Ł. Żel.hotyns, and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” 2019, arXiv:1911.02116.
[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.
[5] C. Dyer, V. Chabane, and N. A. Smith, “A simple, fast, and effective reparameterization of lstm model 2,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol., 2013, pp. 644–648.
[6] S. Eo, C. Park, H. Moon, J. Seo, and H. Lim, “Dealing with the paradox of quality estimation,” in Proc. 4th Workshop Technol. MT Low Resource Lang. (LoResMT), Aug. 2021, pp. 1–10. [Online]. Available: https://aclanthology.org/2021.mtsummit-loresmt.1
[7] S. Eo, C. Park, H. Moon, J. Seo, and H. Lim, “Comparative analysis of current approaches to quality estimation for neural machine translation,” Appl. Sci., vol. 11, no. 14, pp. 6584, Jul. 2021.
[8] S. Eo, C. Park, J. Seo, H. Moon, and H. Lim, “Word-level Korean-English quality estimation,” in Proc. Annu. Conf. Hum. Lang. Technol., 2021, pp. 9–15.
[9] C. Hardmeier, J. Nivre, and J. Tiedemann, “Tree kernels for machine translation quality estimation,” in Proc. 5th Workshop Stat. Mach. Transl., Montreal, QC, Canada, Jun. 2012, pp. 109–113.
[10] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
[11] N. Houlsby, A. Giugiu, S. Jastrzębski, B. Morrone, Q. De Laroussilhe, A. Gensmundo, M. Attarian, and S. Gelly, “Parameter-efficient transfer learning for NLP,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 2790–2799.
[12] F. Kepler, J. Trénous, M. Tresvo, M. Vera, A. Góis, M. Amin Farajian, A. V. Lopes, and A. F. T. Martins, “Unbabel’s participation in the WMT19 translation quality estimation shared task,” 2019, arXiv:1907.10352.
[13] H. Kim and J.-H. Lee, “Recurrent neural network based translation quality estimation,” in Proc. 1st Conf. Mach. Transl.: Shared Task Papers, vol. 2, 2016, pp. 787–792.
[14] H. Kim and S.-H. Na, “Uniformly interpolated balancing for robust prediction in translation quality estimation: A case study of English-Korean translation,” ACM Trans. Asian Low-Resour Lang. Inf. Process., vol. 19, no. 3, pp. 1–27, 2020.
[15] H. Kim, J.-H. Lee, and S.-H. Na, “Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation,” in Proc. 2nd Conf. Mach. Transl., 2017, pp. 562–568.
[16] H. Kim, J. Shin, W. Lee, S. Cho, and J.-H. Lee, “Quality estimation of English-Korean machine translation using neural network based predictor-estimator mode,” Korean Inst. Inf. Scientists Eng., vol. 45, no. 6, pp. 545–553, 2018.
[17] H. Kim, J.-H. Lim, H.-K. Kim, and S.-H. Na, “QE BERT: Bilingual BERT using multi-task learning for neural quality estimation,” in Proc. 4th Conf. Mach. Transl., vol. 3, 2019, pp. 85–89.
[18] P. Koehn, H. Hoang, and A. Birch, “Moses: Open source toolkit for statistical machine translation,” in Proc. 45th Annu. Meeting Assoc. Comput. Linguistics Companion Volume Demo Poster Sessions, 2007, pp. 177–180.
[19] T. Kudo and J. Richardson, “SentencePiece: A simple and language-independent subword tokenizer and detokenizer for neural text processing,” 2018, arXiv:1808.06226.
[20] G. Lample and A. Conneau, “Cross-lingual language model pretraining,” 2019, arXiv:1901.07291.
[21] D. Lee, “Cross-lingual transformers for neural automatic post-editing,” in Proc. 5th Conf. Mach. Transl., 2020, pp. 772–776.
[22] D. Lee, “Two-phase cross-lingual language model fine-tuning for machine translation quality estimation,” in Proc. 5th Conf. Mach. Transl., 2020, pp. 1024–1028.
[23] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” 2019, arXiv:1910.13461.
[24] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, “Multilingual denoising pre-training for neural machine translation,” Trans. Assoc. Comput. Linguistics, vol. 8, pp. 726–742, Dec. 2020.
[25] N.-Q. Luong, B. Lecoutreux, and L. Besacier, “LIG system for WMT13 QE task: Investigating the usefulness of features in word confidence estimation,” in Proc. 8th Workshop Stat. Mach. Transl., 2013, pp. 386–391.
[26] J. Moura, M. Vera, D. van Stigt, F. Kepler, and A. F. Martins, “IST-unbabel participation in the WMT20 quality estimation shared task,” in Proc. 5th Conf. Mach. Transl., 2020, pp. 1029–1036.
[27] C. Park, S. Eo, H. Moon, and H. Lim, “Should we find another model?: Improving neural machine translation performance with ONE-piece tokenization method without model modification,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol., Ind. Papers, 2021, pp. 97–104.
[28] K. Park, J. Lee, S. Jang, and D. Jung, “An empirical study of tokenization strategies for various Korean NLP tasks,” 2020, arXiv:2010.02534.
[29] R. Nath Patel, “Translation quality estimation using recurrent neural network,” 2016, arXiv:1610.04841.
[30] T. Ranasinghe, C. Orasan, and R. Mitkov, “TransQuest at WMT2020: Sentence-level direct assessment,” 2020, arXiv:2010.05318.

[31] T. Ranasinghe, C. Orasan, and R. Mitkov, “An exploratory analysis of multilingual word-level quality estimation with cross-lingual transformers,” 2021, arXiv:2106.00143.

[32] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” 2015, arXiv:1508.07909.

[33] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, “A study of translation edit rate with targeted human annotation,” in Proc. 7th Conf. Assoc. Mach. Transl. Amer., vol. 200, 2006, pp. 223–231.

[34] R. Soricut, N. Bach, and Z. Wang, “The SDL language weaver systems in the WMT12 quality estimation shared task,” in Proc. 7th Workshop Stat. Mach. Transl., 2012, pp. 145–151.

[35] L. Specia, D. Raj, and M. Turchi, “Machine translation evaluation versus quality estimation,” Mach. Transl., vol. 24, no. 1, pp. 39–50, Mar. 2010.

[36] L. Specia, F. Blain, M. Fomicheva, E. Fonseca, V. Chaudhary, F. Guzmán, and A. F. T. Martins, “Findings of the WMT 2021 shared task on quality estimation,” in Proc. 5th Conf. Mach. Transl., Nov. 2020, pp. 743–764. [Online]. Available: https://www.aclweb.org/anthology/2020.wmt-1.79

[37] S. Sun, M. Fomicheva, F. Blain, V. Chaudhary, A. El-Kishky, A. Renduchintala, F. Guzmán, and L. Specia, An Exploratory Study on Multilingual Quality Estimation. Stroudsburg, PA, USA: Association for Computational Linguistics, 2020.

[38] W. L. Taylor, “‘Cloze procedure’: A new tool for measuring readability,” Journalism Quart., vol. 30, no. 4, pp. 415–433, 1953.

[39] M. Wang, H. Yang, H. Shang, D. Wei, J. Guo, L. Lei, Y. Qin, S. Tao, S. Sun, Y. Chen, and L. Li, “HW-TSC’s participation at WMT 2020 quality estimation shared task,” in Proc. 5th Conf. Mach. Transl., 2020, pp. 1056–1061.

[40] T. Wolf et al., “HuggingFace’s transformers: State-of-the-art natural language processing,” 2019, arXiv:1910.03771.

[41] L. Zhou, L. Ding, and K. Takeda, “Zero-shot translation quality estimation with explicit cross-lingual patterns,” 2020, arXiv:2010.04989.

SUGYEONG EO received the B.S. degree in linguistics and cognitive science, language and technology from the Hankuk University of Foreign Studies, Yongin, South Korea, in 2020. She is currently pursuing the Ph.D. degree in computer science and engineering with Korea University, Seoul, South Korea. She is also a part of the Natural Language Processing and Artificial Intelligence Laboratory, under an integrated master’s and Ph.D. course. Her research interests include neural machine translation and quality estimation, where she tries to predict machine translation quality that minimizes human labor.

CHANJUN PARK received the B.S. degree in natural language processing and creative convergence from the Busan University of Foreign Studies, Busan, South Korea, in 2019. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Korea University, Seoul, South Korea. From June 2018 to August 2019, he worked at SYSTRAN as a Research Engineer. He is also working at Upstage as an AI Research Engineer. His research interests include machine translation, grammar error correction, simultaneous speech translation, and deep learning.

HYEONSEOK MOON received the B.S. degree from the Department of Science in Mathematics and Engineering, Korea University, Seoul, South Korea, in 2021, where he is currently pursuing the Ph.D. degree in computer science and engineering. He is also a part of the Natural Language Processing and Artificial Intelligence Laboratory, under an integrated master’s and Ph.D. course. His research interests include natural language processing, neural machine translation, automatic post editing, and parallel corpus filtering.

JAEHYUNG SEO received the B.S. degree from the Department of English Language and Literature, Korea University, Seoul, South Korea, in 2020, where he is currently pursuing the Ph.D. degree in computer science and engineering. He is also a part of the Natural Language Processing and Artificial Intelligence Laboratory, under an integrated master’s and Ph.D. course. His research interests include language generation and decoding strategy, where he attempts to find inspiration from how humans do it and build generative model based on commonsense reasoning.

HEUISEO LIM received the B.S., M.S., and Ph.D. degrees in computer science and engineering from Korea University, Seoul, South Korea, in 1992, 1994, and 1997, respectively. He is currently a Professor with the Department of Computer Science and Engineering, Korea University. His research interests include natural language processing, machine learning, and artificial intelligence.