# PATQUEST: Papago Translation Quality Estimation

**Yujin Baek**[*,†]
Graduate School of AI, KAIST
yujinbaek@kaist.ac.kr

**Zae Myung Kim**[*]
Papago, Naver Corp.
zaemyung.kim@navercorp.com

**Jihyung Moon**
Papago, Naver Corp.
jihyung.moon@navercorp.com

**Hyunjoong Kim**
Papago, Naver Corp.
soy.lovit@navercorp.com

**Eunjeong L. Park**
Papago, Naver Corp.
lucy.park@navercorp.com

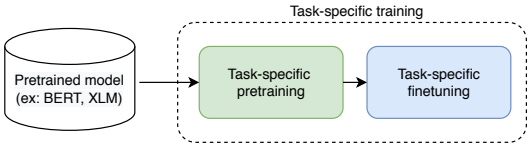
## Abstract

This paper describes the system submitted by Papago team for the quality estimation task at WMT 2020. It proposes two key strategies for quality estimation: (1) task-specific pretraining scheme, and (2) task-specific data augmentation. The former focuses on devising learning signals for pretraining that are closely related to the downstream task. We also present data augmentation techniques that simulate the varying levels of errors that the downstream dataset may contain. Thus, our PATQUEST models are exposed to erroneous translations in both stages of task-specific pretraining and finetuning, effectively enhancing their generalization capability. Our submitted models achieve significant improvement over the baselines for Task 1 (Sentence-Level Direct Assessment; EN-DE only), and Task 3 (Document-Level Score).

## 1 Introduction

With the widespread use of machine translation systems, there is a growing need to evaluate translated results at low-cost. The task of quality estimation (QE) addresses this issue, where the quality of a translation is predicted automatically given the source sentence and its translation. The estimated quality can inform users about the reliability of the translation, or whether it needs to be post-edited.

Previous QE systems generally include pretraining and finetuning steps, where the former step involves masked language modeling (MLM) utilizing large parallel corpora, with the expectation that the models will learn cross-lingual relationships (Kepler et al., 2019; Kim et al., 2019). The models are, in turn, finetuned with task-specific data. However, while the pretraining step involves training data with near-perfect translations, low-quality translations are only introduced during the finetuning step.




| | Task1 (Section 2) | Task3 (Section 3) |
|---|---|---|
| Perturbations | - Omitted word<br>- Word order<br>- Lexical selection<br>- Repeated phrase | - Omitted word (preposition, determiner)<br>- Word order<br>- Lexical selection<br>- Wrong preposition |
| Label | - SentenceBERT<br>- BERTScore<br>- GPT-2 | - Pseudo-MQM |

Figure 1: Overview of our approach for Task 1 and 3.

In this work, we suggest two key strategies that could alleviate this pretrain-finetune discrepancy in QE tasks by: (1) adopting a task-specific pretraining objective which is close to that of the downstream task, and (2) generating abundant task-specific erroneous sentence pairs and their learning signals. Our approach, which is depicted in Figure 1, is motivated from BLEURT (Sellam et al., 2020), where we extend their general approach to the bilingual QE setting. Our submitted systems achieve significant improvements in performance over the baseline systems on WMT20 Shared Tasks for QE (Specia et al., 2020): an absolute gain of $+35.2\%$ in Pearson score for (Task 1) Sentence-Level Direct Assessment (EN-DE), and $+18.4\%$ in Pearson score for (Task 3) Document-Level Score.

## 2 Sentence-Level QE: Direct Assessment

The task of sentence-level QE for direct assessment (DA) involves predicting the perceived quality of the translation given the source and the translated sentences.

Following the footsteps of the previous work

[*] Equal contribution
[†] Work done during internship at Naver Corp.



*Proceedings of the 5th Conference on Machine Translation (WMT)*, pages 991–998
Online, November 19–20, 2020. 

on QE, our sentence-level system also utilizes the pretrained multilingual language models such as BERT (Devlin et al., 2018) and Cross-lingual Language Model (XLM) (Conneau and Lample, 2019). As the size of the training corpus for the QE task is very limited (7K sentence pairs), it is crucial to align these models closely to the task using more data in the form of task-specific pretraining.

As opposed to pretraining the models on parallel corpora using the standard MLM approach, we pretrain the models in a multi-task setting using learning signals and data that are arguably more task-specific similar to Sellam et al. (2020).

### 2.1 Task-Specific Data Augmentation

In order to better align the pretrained models to the QE task, synthetic sentence pairs that contain various types of translation errors are generated from clean parallel corpora[1]. For each target sentence, we generate two perturbed sentences by separately applying one of the four methods described below.

**Omitted Word** We randomly omit at most three words from the target-side, simulating inadequate translations.

**Word Order** Based on the part-of-speech (POS) tag for each word in the target sentence, and predefined sequences of POS patterns, we randomly swap two target words if those words match one of the patterns. The POS patterns can be contiguous, e.g., *adjective-space-noun*, or long-ranged, e.g., *noun-*-adjective*. When none of the patterns are matched, we randomly swap two words.

**Lexical Selection** For each target sentence, we mask out at most three words randomly, and apply mask-filling via a German BERT model from Hugging Face[2]. The purpose of this alteration is to generate fluent but somewhat inadequate target sentences.

**Repeated Phrase** In order to simulate the repetition problem in translations generated by neural machine translation models, we alter the target sentence by adding a repetition of a random phrase within the sentence. The length of the random phrase is at most three tokens.

[1] `Europarl v10` and `News Commentary v15`

[2] `bert-base-german-cased`, https://huggingface.co/transformers/pretrained_models.html

### 2.2 Task-Specific Learning Signals

As the goal of the downstream task is to predict the DA scores which represent the "perceived quality" of the translation, we need to consider pretraining signals that can capture the somewhat subjective notion of "good" and "bad" translations.

Consulting the related works, we prepared the three learning signals:

- SentenceBERT score (Reimers and Gurevych, 2019)
- BERTScore (Zhang et al., 2019), extended to multilingual setting
- Target (German) Language Model (GPT-2, Radford et al. (2019)) score

For each sentence pair in the original bilingual corpora as well as the augmented ones, the three types of learning signals are computed, and later used in the task-specific pretraining.

#### 2.2.1 SentenceBERT Score

For a given sentence, SentenceBERT produces a semantically meaningful sentence embedding that can be compared using a distance metric.

We note that when comparing the distance between two sentence vectors, the Kendall rank correlation coefficient (Kendall, 1938) is computed instead of the cosine similarity measure as the former correlates better with the human judgement, possibly because it produces a more widespread range of scores than the latter especially when the dimension of the sentence vectors is high.

In our experiments, we used the publicly available multilingual SentenceBERT model released from UKPLab[3] that supports 13 languages including English and German.

#### 2.2.2 Multilingual BERTScore

While SentenceBERT score looks at the sentence embedding as a whole, BERTScore computes a similarity score for each token in the pair of sentences. We include BERTScore as one of the learning signals because we feared that the mean-pooling of the BERT-embedded tokens within the SentenceBERT model, while effective in extracting the overall meaning of the sentence, may overlook some of the small semantic details within the sentence.

[3] `distiluse-base-multilingual-cased`, https://github.com/UKPLab/sentence-transformers

However, as the original BERTScore is designed to work in monolingual setting, i.e. evaluating a translation against a reference sentence, it needs to be extended in multilingual setting using a multilingual BERT (mBERT) model. Analogous to the original approach, the multilingual BERTScores can be computed in various ways depending on which side we are computing the maximum similarities from.

In our experiments, we devise a metric where we merge both the source- and target-side maximum similarities between tokens with the corresponding inverse document frequency (IDF) weighting; thus, given a sequence of vectorized source and target tokens, $s$ and $t$, we defined the `mBERTScore` of $s$ and $t$ to be:

$$\frac{S_{s\to t} + S_{t\to s}}{\sum_{s_i \in s} \text{idf}(s_i) + \sum_{t_j \in t} \text{idf}(t_j)}$$

where

$$S_{s\to t} = \sum_{s_i \in s} \text{idf}(s_i) \max_{t_j \in t} \boldsymbol{s_i}^\top \boldsymbol{t_j}$$

$$S_{t\to s} = \sum_{t_j \in t} \text{idf}(t_j) \max_{s_i \in s} \boldsymbol{t_j}^\top \boldsymbol{s_i}$$

#### 2.2.3 Target Language Model Score

While SentenceBERT and multilingual BERTScore can be used as proxies for evaluating the "adequacy" of the translation, empirically, we noticed that they cannot seem to sufficiently represent the "fluency" of translated target sentence. In other words, both metrics may assign high scores to the translated sentence if key source tokens are translated and present in the translation, even when the overall sentence may not be articulate.

To address this issue, the target language model (GPT-2) score is added to the set of learning signals. We simply use the arithmetic mean of the token-level predictions to produce the score for a target sentence. We utilize the pretrained GPT-2 model for German released by Zamia Brain[4].

### 2.3 Model Architecture

We have two stages for task-specific training, i.e. first with the augmented data and the learning signals, and second with the provided QE dataset (ref. Section 2.4). As the output to predict for each stage is different, we utilize the following two types of model architectures.

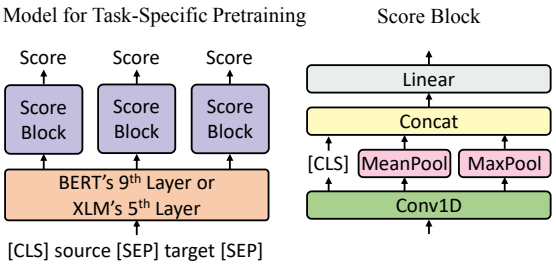



Figure 2: The model architecture (*left*) for the task-specific pretraining using the augmented dataset and learning signals. It consists of three separate Score Blocks (*right*) added on top of the BERT's or XLM's layer.

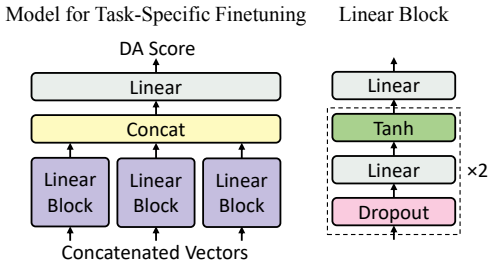



Figure 3: The model architecture (*left*) for the task-specific finetuning using the provided QE dataset. For each concatenated vector computed within each Score Block (c.f. Fig. 2.), a Linear Block (*right*) is added on top of it. The results from the Linear Blocks are concatenated and used to produce the final DA score.

#### 2.3.1 Model for Task-Specific Pretraining

On top of the specific layer of the pretrained mBERT or XLM models, we attach a series of layers called "Score Block" for each type of learning signal as depicted in Figure 2. We utilize the 9th and 5th layer of the BERT and XLM models, respectively, as these layers are reported to be more semantically relevant (Jawahar et al., 2019; Zhang et al., 2019).

In addition to using the vector representation of the `[CLS]` token, utilizing the mean-pooled and max-pooled vectors from all tokens further improved the performance.

#### 2.3.2 Model for Task-Specific Finetuning

Once the task-specific pretraining is completed, we begin the finetuning by adding layers above the concatenation layer within each Score Block, as

[4] `gpt2-german-345M-r20191119`, http://zamia-speech.org/brain

shown in Figure 3. Thus, we have three concatenated vectors being fed to three "Linear Blocks" separately, whose purpose is to reduce the dimensions of the hidden representation, preparing it for the final regression layer.

We note that applying dropout (Srivastava et al., 2014) to these linear layers helps with the performance.

### 2.4 Task-Specific Training

We experiment with three different types of pretrained models: mBERT[5], XLM trained with MLM (XLM-MLM)[6], and XLM trained with causal language modeling (XLM-CLM)[7]. All of the pretrained models are available at Hugging Face.

#### 2.4.1 Task-Specific Pretraining (TSP)

As the size of the provided QE dataset is small, we make use of the existing parallel data as well as the error-induced synthetic data. For the EN-DE bilingual dataset, we select a subset from this year's training corpora for WMT News Translation Task, summing to just under 10M sentence pairs; for the synthetic dataset, the size is 3.4M.

Given the concatenated source and target sentences as an input, the model for TSP is trained to predict the three types of learning signals in a multi-task setting by minimizing the sum of the mean squared error losses for each signal (ref. Figure 2).

#### 2.4.2 Task-Specific Finetuning (TSF)

Once the model is trained with the augmented data, its parameters are loaded to the model for TSF (ref. Figure 3), and finetuned using the QE dataset. This time, the model learns to predict the mean z-normalized DA score.

## 3 Document-Level QE: MQM Scoring

Given a source and its translated document, this task involves identifying translation errors and estimating the translation quality of the document based on the taxonomy of the Multidimensional Quality Metrics (MQM)[8]. With the pre-defined MQM taxonomy, human annotators assess whether the translation satisfies the specifications, and from these annotations, an MQM score is obtained. In this work, we focus on building a system that predicts the MQM score for a given pair of source and translated document.

The major difficulty that we encountered in this task was the lack of training data. As the amount of provided data is limited (8,591 sentence pairs), a model that is solely finetuned on this small-scale data was not capable enough to differentiate sentences with varying level of errors.

To address this issue, we propose simple yet effective methods for task-specific data augmentation, and task-specific training framework[9].

### 3.1 Task-Specific Data Augmentation

We generate erroneous sentence pairs and their pseudo-MQM scores from Europarl and QE training corpus in accordance with the MQM taxonomy.

#### 3.1.1 Generating Erroneous Sentence Pairs

Out of the 45 error categories specified in QE annotations, we select five frequent categories for which we can automatically perturb the target-side of the parallel corpus at little cost. More details on our data augmentation technique for each category are provided below.

**Omitted Preposition** We introduce an error into the target-side of a sentence pair by randomly omitting one of the French prepositions that exist in the sentence.

**Omitted Determiner** The same process is done for French determiners as for prepositions.

**Wrong Preposition** We replace a French preposition with another one. When more than one candidate exists, we choose one at random.

**Word Order** We exploit grammatical pattern that most descriptive adjectives go after the noun in French sentences (unlike English ones). Using an in-house French POS tagger, we identify post-nominal adjectives and place them in front of the corresponding nouns so that they are now pre-nominal.

**Lexical Selection** We mask-out target tokens at random positions, and substitute them with tokens predicted by the Camembert language model (Martin et al., 2020).

[5] `bert-base-multilingual-cased`
[6] `xlm-mlm-ende-1024`
[7] `xlm-clm-ende-1024`
[8] `http://www.qt21.eu/mqm-definition`
[9] The code will be available at `https://github.com/naver/PATQUEST`.

| Error name | Sentence | Length | Total error severity | Pseudo MQM |
|---|---|---|---|---|
| Original sentence | Vous avez souhaité un débat à ce sujet dans les prochains jours, au cours de cette période de session. | 21 | 0 | 100.0 |
| (1) Wrong Preposition | Vous avez souhaité un débat à ce sujet *chez* les prochains jours, au cours de cette période de session. | 21 | 5 | 76.2 |
| (2) Omit Determiner | Vous avez souhaité ~~un~~ débat à ce sujet dans les prochains jours, au cours de cette période de session. | 21 | 5 | 76.2 |
| (1)+(2) | Vous avez souhaité ~~un~~ débat à ce sujet *chez* les prochains jours, au cours de cette période de session. | 20 | 10 | 52.4 |
| Original sentence | Cela placerait l'UE dans une situation délicate vis-à-vis de ces pays et de la communauté internationale. | 23 | 0 | 100.0 |
| (1) Word Order | Cela placerait l'UE dans une situation délicate vis-à-vis de ces pays et de la *internationale communauté*. | 23 | 5 | 78.3 |
| (2) Lexical Selection | Cela placerait l'UE dans une situation *inconfortable* vis-à-vis de ces pays et de la communauté internationale. | 23 | 5 | 78.3 |
| (1)+(2) | Cela placerait l'UE dans une situation *inconfortable* vis-à-vis de ces pays et de la *internationale communauté*. | 23 | 10 | 56.5 |

Table 1: Examples of erroneous sentence pairs generated from the Europarl corpus.

| Error name | Sentence | Length | Total error severity | Pseudo MQM |
|---|---|---|---|---|
| Original sentence | son travail a été présenté dans le washington post, *quotidien bonbons*, washingtonian, fit yoga et journal *d*'yoga. | 23 | 15 | 34.8 |
| (1) Wrong Preposition | son travail a été présenté *pour* le washington post, *quotidien bonbons*, washingtonian, fit yoga et journal *d*'yoga. | 23 | 20 | 13.0 |
| (2) Omit Determiner | son travail a été présenté dans ~~le~~ washington post, *quotidien bonbons*, washingtonian, fit yoga et journal *d*'yoga. | 22 | 20 | 9.1 |
| (1)+(2) | son travail a été présenté *pour* ~~le~~ washington post, *quotidien bonbons*, washingtonian, fit yoga et journal *d*'yoga. | 22 | 25 | -13.6 |
| Original sentence | Brûleur deux *plaque* de cuisson anti-adhésive de Coghlan | 10 | 5 | 50.0 |
| (1) Omit Preposition | Brûleur deux *plaque* de cuisson anti-adhésive ~~de~~ Coghlan | 9 | 10 | -11.1 |

Table 2: Examples of erroneous sentence pairs generated from the WMT20 QE corpus.

#### 3.1.2 Task-Specific Learning Signal

Once we introduce different types of errors into the target-side sentences, the next step is to obtain pseudo-MQM scores for the altered sentence pairs. Two key elements for computing MQM score are the length of a text, and its total error severity as follows:

$$\text{Pseudo-MQM} = 100(1 - \frac{5.0 * n_{error} + S}{N})$$

where $N$ indicates the length of given target sentence and $n_{error}$ denotes the number of errors introduced in it. We assign 5.0, the most frequent severity, to each perturbation that we make. If an error severity score, $S$, is assigned to the sentence by human annotators, we add this score to compute the total error severity score.

### 3.2 Model Architecture

We use pretrained mBERT or XLM[10] as initial parameters. The concatenation of a source sentence and its corresponding target sentence with special symbol tokens is taken as input: `[CLS] source [SEP] target [SEP]`.

We experiment with two strategies for obtaining sentence embeddings. First, we feed a hidden state vector corresponding to `[CLS]` token ($h_{[CLS]}$) to a linear layer to compute a sentence-level MQM prediction of $\hat{y}$:

$$\hat{y} = W h_{[CLS]} + b$$

where $W$ and $b$ are the weight matrix and bias vector of the linear layer, respectively. For the other method, we use the concatenation of a mean-pooled source representation ($s \in \mathbb{R}^n$), mean-pooled target representation ($t \in \mathbb{R}^n$) and their element-wise differences ($|s - t| \in \mathbb{R}^n$) in an attempt to enlarge the model capacity:

$$\hat{y} = W \cdot \text{ReLU}(W_r(s, t, |s - t|) + b_r) + b$$

where $W_r \in \mathbb{R}^{3n \times n}$ and $b_r$ are the weight matrix and bias vector of an intermediate dimension-reducing layer, respectively, and $n$ denotes the dimension of hidden vectors. $W$ and $b$ are the weight matrix and bias vector of the final linear layer.

### 3.3 Task-Specific Training

We suggest that the pretraining objective should be similar to that of the downstream task in order to mitigate the pretrain-finetune discrepancy (Yang et al., 2019), and fully leverage the erroneous sentence pairs that we generated. For this task, both phases minimize the mean-squared loss function: $l = \frac{1}{K} \sum_{k=1}^{K} \|y_k - \hat{y}\|^2$.

#### 3.3.1 Task-Specific Pretraining (TSP)

We utilize Europarl parallel corpus (English-French) to pretrain our submitted models[11]. To acquire high quality data, we carried out the following filtering processes: (1) language detection (filtering out non-English sentences in the source-side, and non-French sentences in the target-side), (2) length ratio filtering (eliminating sentence pairs with length ratio greater than 1.8).

[10] `xlm-mlm-enfr-1024`

[11] We perform TSP after bringing pretrained parameters of language models as initial weights.

We assume that the remaining sentence pairs do not contain any translation error. Therefore, we assign the total error severity score of zero to these pairs before the augmentation.

About 15.2 million examples[12] are generated with the above-mentioned data augmentation techniques. The detailed examples are provided in Table 1.

#### 3.3.2 Task-Specific Finetuning (TSF)

The next step is to finetune our model using the augmented QE train data. Unlike Europarl corpus, we can fully leverage the MQM scores originally assigned to the QE training dataset. We found that performing the data augmentation with three categories (*Omitted Determiner*, *Omitted Preposition*, and *Wrong Preposition*) effectively improves the performance. The original QE training sentence pairs represent about 5% of 169,997 sentence pairs obtained from the data augmentation. We also provide the augmented examples for QE training data in Table 2.

Since the learning objective is identical to that of the pretraining phase, we can simply train the same model with the augmented downstream task data.

### 3.4 Document-Level MQM Score

We specify that the models are trained at sentence-level, learning to predict the non-truncated version of MQM scores which could take a range between negative infinity and 100; this is to avoid potential information loss that could arise from the truncation.

Given a document, the document-level MQM score is computed from its sentence-level MQM predictions in a closed form. Afterwards, we truncate negative values to zero.

## 4 Experimental Results

### 4.1 Sentence-Level Task

Table 3 shows the Pearson correlation coefficient between the predicted z-normalized DA scores and the reference scores on the development set. We note that the number of parameters for PATQUEST-mBERT (724M) is greater than that of PATQUEST-XLM (616M) models, resulting in the difference in the correlation scores. Nevertheless, computing the arithmetic mean of the scores produced by these three models improves the performance (PATQUEST-ensemble).

| Model | Pearson's r ↑ |
|---|---|
| PATQUEST-mBERT | 0.486 |
| PATQUEST-XLM-MLM | 0.450 |
| PATQUEST-XLM-CLM | 0.452 |
| PATQUEST-ensemble | **0.501** |

Table 3: Results on the *development* set for Task 1 EN-DE.

| Model | Pearson's r ↑ | MAE ↓ | RMSE ↓ |
|---|---|---|---|
| Baseline | 0.146 | 0.679 | 0.967 |
| PATQUEST-mBERT w/o synth. data | 0.429 | 0.462 | 0.632 |
| PATQUEST-ensemble w/o synth. data | 0.457 | 0.464 | 0.640 |
| PATQUEST-ensemble | **0.498** | **0.454** | **0.637** |

Table 4: Submission results on the *test* set for Task 1 EN-DE.

The final result on the QE test set is shown in Table 4. We observe that finetuning the model with the additional error-induced synthetic data improves the performance as well as ensembling the models.

Our final submitted system (PATQUEST-ensemble) finished 4th out of the 15 submitted systems[13] in the final ranking of the sentence-level QE task for English-German. In order to train a generally applicable QE system, we did not make use of the data such as internal information from the NMT models and in-domain Wikipedia texts that could be extracted from the provided Wikipedia titles.

### 4.2 Document-Level Task

The validation results on development set are shown in Table 5. Both PATQUEST-mBERT and PATQUEST-XLM models use representations from [CLS] token. We build another two models, PATQUEST-mBERT variant 1 and 2, using the concatenations of mean-pooled source representations, mean-pooled target representations, and their element-wise differences.

Table 6 shows the test results of our submitted PATQUEST models. For PATQUEST-ensemble, we compute an average from the four models enumerated in Table 5.

In Table 7, the effectiveness of our training scheme and data augmentation techniques is illustrated via an ablation study. Note that "Pretrained mBERT (A)" in the table refers to the mBERT

[12]The size of the original Europarl English-French parallel corpus is about 2M sentence pairs.

[13]Excluding the disqualified team.

| Model | Pearson's r ↑ | MAE ↓ | RMSE ↓ |
|---|---|---|---|
| PATQUEST-mBERT | 0.431 | 14.401 | 22.330 |
| PATQUEST-mBERT variant 1 | 0.406 | 14.418 | 22.872 |
| PATQUEST-mBERT variant 2 | 0.380 | 14.909 | 23.215 |
| PATQUEST-XLM | 0.374 | 16.245 | 23.647 |

Table 5: Results on the *development* set of WMT20 document-level task.

| Model | Pearson's r ↑ | MAE ↓ | RMSE ↓ |
|---|---|---|---|
| Baseline | 0.389 | 19.939 | 26.608 |
| PATQUEST-mBERT | 0.529 | 16.214 | 24.437 |
| PATQUEST-XLM | 0.546 | 15.821 | 23.846 |
| PATQUEST-ensemble | **0.573** | **15.611** | **23.327** |

Table 6: Submission results of PATQUEST models on the *test* set of WMT20 document-level task.

model that is finetuned on the original QE data without any task-specific training. Both TSP and TSF enhance the generalization ability of model. Note that the mBERT model trained via TSP and TSF, "A + TSP + TSF", is the same model as PATQUEST-mBERT which itself achieves a significant improvement over the baselines as shown in Table 6.

Our final system (PATQUEST-ensemble) submitted for the document-level QE task, came 1st out of the three submitted systems[14]. Similar to our sentence-level system, our document-level system also did not utilize any internal information from the NMT models and in-domain Wikipedia data tailored to the benchmark.

## 5 Conclusion

In this paper, we present a task-specific pretraining scheme for the QE task. Our pretraining objective is devised so that it is closely related (Task 1) or identical (Task 3) to the finetuning objective. In addition, the models are exposed to abundant amount of error-induced translations generated from large parallel corpora, effectively alleviating the issue of data scarcity.

Our proposed models yield significant improvement over the baseline systems for the two tasks.

[14]Excluding the disqualified team

| Model | Pearson's r ↑ | MAE ↓ | RMSE ↓ |
|---|---|---|---|
| Pretrained mBERT (A) | 0.263 | 16.146 | 23.090 |
| A + TSF | 0.341 (+ 0.078) | 15.302 | 23.749 |
| A + TSP | 0.375 (+ 0.112) | 15.496 | 23.444 |
| A + TSP + TSF | 0.431 (+ 0.168) | 14.401 | 22.330 |

Table 7: Results on the *development* set of WMT20 document-level task adding up key components of our model.

## Acknowledgments

Authors would like to thank Stéphane Clinchant, Vassilina Nikoulina, and Jaesong Lee for the insightful discussions, and Papago team members for offering the fruitful feedback. We would also like to extend our gratitude to Won Ik Cho for coming up with the awesome name for our system.

## References

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M Amin Farajian, António V Lopes, and André FT Martins. 2019. Unbabel's participation in the wmt19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 78–84.

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 (Volume 3: Shared Task Papers, Day 2)*, pages 85–89.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.