Two-Phase Cross-Lingual Language Model Fine-Tuning for Machine Translation Quality Estimation

Dongjun Lee
Bering Lab, Republic of Korea
djlee@beringlab.com

Abstract

In this paper, we describe the Bering Lab's submission to the WMT 2020 Shared Task on Quality Estimation (QE). For word-level and sentence-level translation quality estimation, we fine-tune XLM-RoBERTa, the state-of-the-art cross-lingual language model, with a few additional parameters. Model training consists of two phases. We first pre-train our model on a huge artificially generated QE dataset, and then we fine-tune the model with a human-labeled dataset. When evaluated on the WMT 2020 English-German QE test set, our systems achieve the best result on the target-side of word-level QE and the second best results on the source-side of word-level QE and sentence-level QE among all submissions.

1 Introduction

Machine translation quality estimation (QE) is the task of estimating the quality of machine-translated (MT) output given just the source text at various granularity levels (word, sentence, and document) (Fonseca et al., 2019). Word-level QE can be divided into target-side and source-side tasks. On the target-side, the goal is to predict whether each word in the MT sentence is `OK` or `BAD` and whether there are missing words between each word. The goal on the source-side is to predict whether each word in the source sentence is correctly translated or not. On the other hand, sentence-level QE aims to predict the Human Translation Error Rate (HTER) (Snover et al., 2006) of the MT sentence, which measures the required amount of human editing to fix the MT sentence.

In this paper, we propose a cross-lingual language model fine-tuning approach with a few additional parameters for word-level and sentence-level QE. As a pre-trained cross-lingual language model, we use XLM-RoBERTa (XLM-R) (Conneau et al., 2019), which shows state-of-the-art performance for a wide range of cross-lingual transfer tasks. In addition, since labeling the QE dataset requires a large amount of human labor, we generate and utilize a huge artificial QE dataset to improve the performance of our model. Our contributions are summarized as follows.

- We propose an XLM-R-based neural network architecture for the QE. Our model can be jointly trained for both word-level and sentence-level QE.
- We generate a huge artificial QE dataset based on a parallel corpus with OpenNMT-py (Klein et al., 2017) and the TER tool (Snover et al., 2006).
- We train our model in two phases. First, we train our model with a huge artificially generated dataset. Then, we fine-tune the model with a human-labeled dataset.

In the experiment using the WMT 2020 English-German word-level QE test set, we achieve an MCC of 0.597 and 0.454 for the target-side and source-side, respectively, showing the best and second best performance among all submitted systems, respectively. For the sentence-level QE test set, we achieve a Pearson correlation of 0.723, which ranks second among all submissions.

2 Methodology

We fine-tune XLM-R (Conneau et al., 2019) with a few additional parameters for sentence-level and word-level QE as described in Figure 1. We train our model in two phases: 1) pre-training with a huge artificial dataset and 2) fine-tuning with a human-labeled dataset.

2.1 Input Representation

We follow the tokenization and input representation methods of XLM-R. A source sentence and

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1024–1028
Online, November 19–20, 2020.

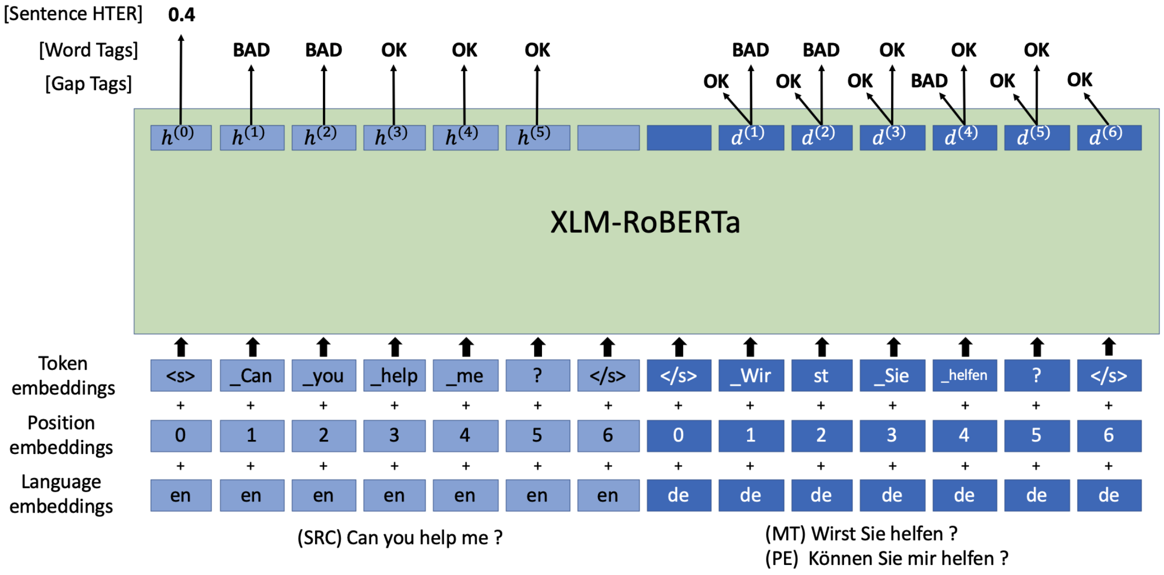

Figure 1: The XLM-R-based neural network architecture for word-level and sentence-level QE.

the corresponding MT sentence are tokenized with the same BPE model (Sennrich et al., 2016) that is trained based on shared vocabulary through languages. The input of the XLM-R model is a concatenated sequence of source tokens and MT tokens with special tokens (<s>, </s>) as follows:

<s> src_1, ..., $src_{|S|}$ </s> </s> mt_1, ..., $mt_{|T|}$ </s>

2.2 Sentence-level QE

For sentence-level QE, we use the final hidden vector $h^{(0)} \in \mathbb{R}^H$ of XML-R corresponding to the first input token (<s>) as the pooled representation. We use two linear layers with tanh activation to predict sentence-level HTER as follows:

$$r = W_s h^{(0)} + b_0 \quad (1)$$

$$y_{sent} = w_s^T tanh(r) + b_1 \quad (2)$$

where $W_s \in \mathbb{R}^{H \times H}$, $w_s \in \mathbb{R}^H$, $b_0 \in \mathbb{R}^H$ and $b_1 \in \mathbb{R}^1$ are trainable parameters and H is the dimension of hidden states.

The loss function L_{sent} is the mean squared error between y_{sent} and the true HTER $\hat{y}_{sent}$.

$$L_{sent} = MSE(y_{sent}, \hat{y}_{sent}) \quad (3)$$

2.3 Word-level QE

Word-level QE consists of two parts: the source-side and target-side. On the source-side, we predict whether each token in the source sentence is translated correctly or not. On the target-side, we predict whether each token in the MT sentence is OK or BAD, in addition to whether there are missing words between each word.

Source-side QE For source-side QE, we use the final hidden vector $h^{(i)} \in \mathbb{R}^H$ of XLM-R corresponding to each token in the source sentence. We introduce a linear layer and sigmoid activation to predict the probability that each token is BAD as follows:

$$P_{src}^{(i)} = sigmoid(w_o^T h^{(i)}), i \in (1, .., |S|) \quad (4)$$

where $w_o \in \mathbb{R}^H$ is a trainable parameter and $|S|$ is the number of tokens in the source sentence.

The loss function L_{src} is the binary cross entropy with an additional weight c for BAD examples as follows:

$$L_{src} = \frac{1}{|S|} \sum_{i=1}^{|S|} c\hat{y}_{src}^{(i)} \log P_{src}^{(i)} + (1 - \hat{y}_{src}^{(i)}) \log(1 - P_{src}^{(i)}) \quad (5)$$

Target-side QE For the target-side QE, we use the final hidden vector $d^{(i)} \in \mathbb{R}^H$ of XLM-R corresponding to each token in the MT sentence, including the last </s> token. We introduce two separate binary classification layers to predict the probability that each token in MT sentence is BAD as follows:

$$P_{tgt\_word}^{(i)} = sigmoid(w_w^T d^{(i)}), i \in (1, .., |T|) \quad (6)$$

and the probability that missing words exist before each token as follows:

$$P^{(i)}_{tgt\_gap} = sigmoid(w_g^T d^{(i)}), i \in (1, .., |T| + 1) \tag{7}$$

where $w_w, w_g \in \mathbb{R}^H$ are trainable parameters and $|T|$ is the number of tokens in the machine translated sentence.

The loss function for target-side QE L_{tgt} is the sum of the binary cross entropy for word L_{tgt_word} and gap L_{tgt_gap} that are defined in the same manner as Eq. (5).

$$L_{tgt} = L_{tgt\_word} + L_{tgt\_gap} \tag{8}$$

2.4 Pre-Training on Artificial Dataset

Building the Artificial Dataset Labeling data for QE requires the triplets of source sentences, machine-translated (MT) sentences, and human post-edited (PE) sentences. Since huge costs are required to achieve PE sentences, we use a parallel corpus that includes only source sentences and target sentences to build artificial triplets following the ideas from Negri et al. (2018).

First, we split the parallel corpus into a training set and test set. We train an NMT model with the training set and use the test set to generate artificial triplets. We generate MT sentences based on the trained NMT model and we use the target sentences of the parallel corpus as PE sentences. We repeat this process with different data splits to build huge artificial triplets. Finally, we use the TER tool[1] (Snover et al., 2006) to annotate sentence-level HTER scores and word-level tags for the MT sentences. We do not annotate source-side word-level tags in this work as it additionally requires word alignment between source sentences and MT sentences.

Pre-training QE Model We first pre-train our QE model with only the artificial dataset. In the pre-training step, we jointly train sentence-level QE and target-side word-level QE on a single model. The loss function for the pre-training step L_{pre_train} is the sum of the loss for sentence-level QE and target-side word-level QE.

$$L_{pre\_train} = L_{sent} + L_{tgt} \tag{9}$$

Since our artificial dataset does not include source-side word-level tags, we do not include the training objective for source-side word-level QE in the pre-training step.

2.5 Fine-Tuning on Human-Labeled Dataset

After the pre-training, we fine-tune the model with only a human-labeled dataset. Unlike the pre-training step, each QE model (sentence-level, source-side and target-side of word-level) is trained separately in the fine-tuning step.

For the sentence-level and target-side of word-level QE models, all the parameters are initialized with trained weights from the pre-training step. However, since our pre-trained model does not include source-side word-level QE, we randomly initialize the weight of a source-side specific parameter (w_o in Eq. (4)) and the rest of the parameters are initialized with weights from the pre-trained model.

2.6 Ensemble

For the sentence-level ensemble, we average the HTER prediction of multiple models. For the word-level, we use the majority voting ensemble.

3 Experiments

3.1 Experimental Setup

We evaluate our model with WMT 2020 English-German QE dataset.[2] For the sentence-level QE evaluation, we use the Pearson correlation for sentence-level HTER prediction. For the word-level QE evaluation, we use the Matthews correlation coefficient (MCC) for both the target-side and source-side.

To generate an artificial dataset for pre-training (§2.4), we use the English-German parallel corpus provided by the shared task that consists of 23,440,059 pairs. We use 90% of the pairs to train a Transformer-based (Vaswani et al., 2017) NMT model with OpenNMT-py (Klein et al., 2017) and the rest of the pairs are used to generate artificial triplets. As a result of running the process five times with different data splits, we achieve 11,720,029 artificial triplets.

For the fine-tuning, we use only the official QE dataset that consists of 7,000 triplets as a human-labeled dataset.

3.2 Model Configuration

We use XLM-R-Large (Conneau et al., 2019) as a pre-trained cross-lingual language model. For pre-training with the artificial dataset, we use the Adam optimizer (Kingma and Ba, 2014) with a

[1] http://www.cs.umd.edu/~snover/tercom/

[2] http://www.statmt.org/wmt20/quality-estimation-task.html

Systems	Pearson↑	Target-side MCC↑	Source-side MCC↑
Ours	**0.715**	**0.591**	**0.464**
-ensemble	0.712	0.586	0.457
-ensemble -pre-train	0.591	0.476	0.365
-ensemble -fine-tune	0.424	0.378	-

Table 1: Ablation analysis for sentence-level and word-level QE on the WMT 2020 English-German QE *dev* set. Since our pre-training step does not include source-side word-level QE, we do not measure the source-side MCC for the pre-trained only model.

Systems	Pearson↑	MAE↓	RMSE↓
HW-TSC	**0.758**	**0.099**	**0.133**
Ours (Bering Lab)	0.723	0.107	0.140
NiuTrans	0.649	0.123	0.154
IST and Unbabel	0.633	0.137	0.178
NJUNLP	0.618	0.129	0.160
Baseline	0.392	0.150	0.190

Table 2: Top-5 and baseline systems from the official result for the sentence-level QE on the WMT 2020 English-German QE shared task.

Systems	Target-side MCC↑	Source-side MCC↑
Ours (Bering Lab)	**0.597**	0.454
HW-TSC	0.583	**0.523**
NiuTrans	0.500	0.347
NICT Kyoto	0.485	0.353
IST and Unbabel	0.465	0.349
Baseline	0.358	0.266

Table 3: Top-5 and baseline systems from the official result for the word-level QE on the WMT 2020 English-German QE shared task.

learning rate of 5e-6, and a batch size of 8 for 2 epochs. Additionally, we use dropout (Hinton et al., 2012) with a rate of 0.1 for the regularization. For word-level QE, we use a weight of 3.0 on the BAD class (c). For fine-tuning with the human-labeled dataset, we follow the same hyperparameters as the pre-training step but for 5 epochs with early stopping. For the ensembling, we train five models with different random seeds.

3.3 Experimental Result

Table 1 shows the result of ablation analysis for sentence-level and word-level QE on the *dev* set. We conduct an ablation analysis of three aspects: 1) without an ensemble, 2) without pre-training with artificially generated dataset, 3) without fine-tuning with human-labeled dataset. When our model is trained with only the human-labeled dataset, Pearson correlation, target-side MCC and source-side MCC drop by 0.12, 0.11, and 0.09, respectively. This result demonstrates that pre-training with the artificial dataset significantly improves performance for both sentence-level and word-level QE. When our model is trained with only the artificial dataset, Pearson correlation and target side MCC drop by 0.29 and 0.21, respectively. This result shows that fine-tuning with a human-labeled dataset is essential for our performance.

Table 2 and 3 shows the official results for sentence-level and word-level QE for the WMT 2020 QE shared task. For both sentence-level and word-level QE, our systems significantly outperformed the official baseline systems (Kepler et al., 2019). Moreover, we achieve the best result on the target side of word-level QE among all submitted systems. We also achieve the second best results on the source side of word-level QE and sentence-level QE.

4 Conclusion

This paper describes Bering Lab's submissions to the WMT 2020 QE shared task. We propose a two-phase cross-lingual language model fine-tuning approach for word-level and sentence-level translation quality estimation. The experimental results show that pre-training with an artificially generated dataset significantly improves performance for both tasks. Overall, our submitted systems achieve the best result on the target side of word-level QE and the second best results on the source side of word-level QE and the sentence-level QE among all submissions.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Erick Fonseca, Lisa Yankovskaya, André FT Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the wmt 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

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

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

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

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 (LREC 2018)*.

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 (Volume 1: Long Papers)*, pages 1715–1725.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

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