SOURCE: SOURce-Conditional Elmo-style Model for Machine Translation Quality Estimation

Junpei Zhou[*] **Zhisong Zhang**[*] **Zecong Hu**[*]
Language Technologies Institute
Carnegie Mellon University
{junpeiz, zhisongz, zeconghu}@andrew.cmu.edu

Abstract

Quality estimation (QE) of machine translation (MT) systems is a task of growing importance. It reduces the cost of post-editing, allowing machine-translated text to be used in formal occasions. In this work, we describe our submission system in WMT 2019 sentence-level QE task. We mainly explore the utilization of pre-trained translation models in QE and adopt a bi-directional translation-like strategy. The strategy is similar to ELMo, but additionally conditions on source sentences. Experiments on WMT QE dataset show that our strategy, which makes the pre-training slightly harder, can bring improvements for QE. In WMT-2019 QE task, our system ranked in the second place on En-De NMT dataset and the third place on En-Ru NMT dataset.

1 Introduction

The quality of machine translation systems have been significantly improved over the past few years (Chatterjee et al., 2018), especially with the development of neural machine translation (NMT) models (Sutskever et al., 2014; Bahdanau et al., 2014). Despite such inspiring improvements, some machine translated texts are still error-prone and unreliable compared to those by professional humans. It is often desirable to have human experts perform post-editing on machine-translated text to achieve a balance between cost and correctness. Correspondingly, we may also want to develop automatic quality estimation systems to judge the quality of machine translation outputs, leading to the development of the Machine Translation Quality Estimation task. The task of QE aims to evaluate the output of a machine translation system without access to reference translations. It would allow human experts to concentrate on translations that are estimated of low-quality, further reducing post-editing cost.

In this work, we focus on sentence-level QE and describe our submission to the WMT 2019 QE task. Sentence-level QE aims to predict a score for the entire source–translation pair that indicates the effort required for further post-editing. The goals of the task are two-fold: 1) to predict the required post-editing cost, measured in HTER (Snover et al., 2006); 2) to rank all sentence pairs in descending translation quality.

In previous works, including the participating systems in previous WMT shared tasks, there have been various methods to tackle this problem. Traditional linear models are based on handcrafted features, while recent state-of-the-art systems adopt end-to-end neural models (Kim and Lee, 2016; Wang et al., 2018). The neural systems are usually composed of two modules: the bottom part is an MT-like source–target encoding model pre-trained with large parallel corpora, stacked with a top-level QE scorer based on the neural features extracted by the bottom model. Especially, Wang et al. (2018) adopted the “Bilingual Expert” model (Fan et al., 2018) for pre-training the bottom model and obtained several best results in WMT 2018. In this work, we improve the “Bilingual Expert” model with a SOURce-Conditional ELMo-style (SOURCE) strategy: instead of predicting target words based on contexts from both sides, we train two conditioned language (translation) models, each restricted to context from one side only. This harder setting may force the model to condition more on the source. Experiments show that this strategy can bring improvements for QE.

[*] equal contribution

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 3: Shared Task Papers (Day 2) pages 108–113
Florence, Italy, August 1-2, 2019. ©2019 Association for Computational Linguistics

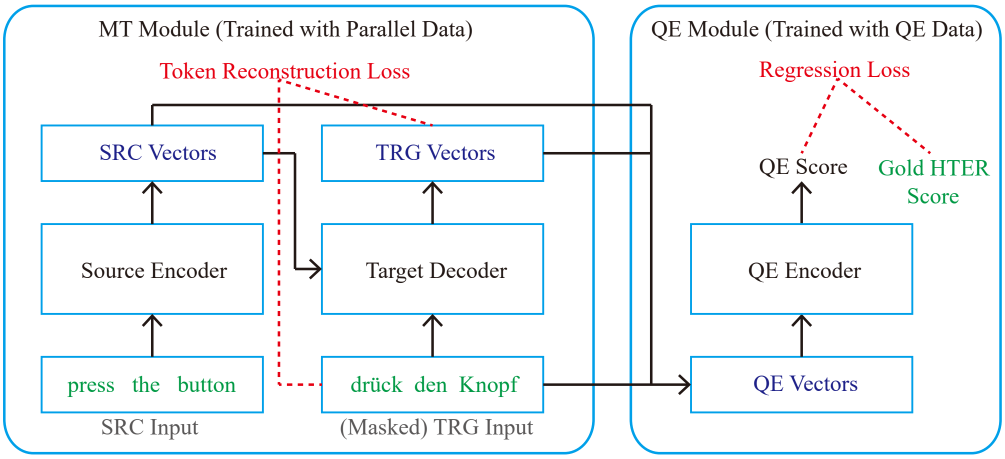

Figure 1: The architecture of our QE system, which consists of two modules: 1) the MT Module encodes the bilingual information and can be pre-trained with large parallel data, 2) the QE Module adopts the source and target representations from the MT Module and further encodes those information followed by a final linear layer for QE scoring.

2 System

2.1 Basic Framework

We follow previous works and adopt the end-to-end styled model for the QE scoring task. The overall system architecture is shown in Figure 1. The system consists of two components: 1) a pre-trained MT module which learns the representations of the source and target sentences, 2) a QE scorer which takes the representations from the MT module as inputs and predicts the translation quality score.

The MT module is pre-trained on large parallel corpus. It is trained to predict each token in the translated sentence by using the information in source sentence and tokens in the translated sentence. Details of the model will be described in Section 2.2.

In the QE scorer module, the problem can be cast as a regression task, where the QE score is predicted given the source and target sentences. The original inputs are encoded by the pre-trained MT module, whose outputs are taken as input features for this module. We basically follow the model architecture of Wang et al. (2018). For each token, a quality vector is formed as:

$$q_j = \text{Concat}(\overleftarrow{z_j}, \overrightarrow{z_j}, e^t_{j-1}, e^t_{j+1}, f^{mm}_j), \qquad (1)$$

where $\overleftarrow{z_j}, \overrightarrow{z_j}$ are state vectors produced by the bi-directional Transformer, and e^t_{j-1}, e^t_{j+1} are embedding vectors. The “mismatching feature” f^{mm}_j is formed by extracting the score corresponding to y_j, the highest score in the distribution, their difference, and an indicator of whether y_j has the highest score. After this, the quality vectors are viewed as another sequence and encoded by the Bi-LSTM/Transformer Quality Estimator to predict the QE score. The loss function for training is mean squared error which is typical for regression tasks.

2.2 Pre-trained Translation Models

Bilingual Expert We start with a short description for the model of Wang et al. (2018). The model can be seen as a token-level reconstruction-styled translation model: each target word y_j is predicted given a source sentence and all other target words $\{\ldots, y_{j-1}, y_{j+1}, \ldots\}$. This setting is different to the traditional MT scenario where only previous target words can be seen. The model uses the encoder–decoder architecture. An encoder is applied over the source tokens to obtain the contextual representations of the source sentence. A bidirectional pair of decoders (one forward and one backward) are adopted to encode the target translation sentence, while conditioning on the source sentence via attention mechanism. Formally, for source tokens $\{x_1, \ldots, x_{m_s}\}$ and translation tokens $\{y_1, \ldots, y_{m_t}\}$, the forward and backward target representations $\{\overrightarrow{z_1}, \ldots, \overrightarrow{z_{m_t}}\}$ and

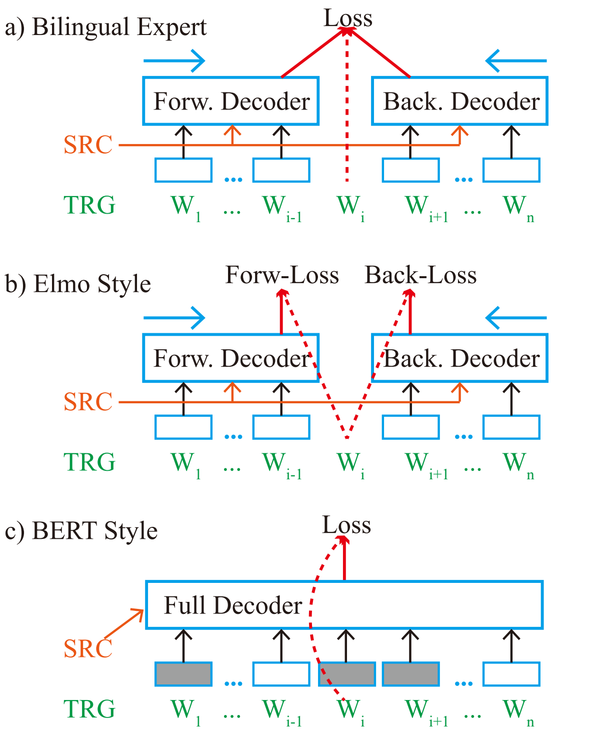

Figure 2: Illustration of reconstruction loss for the token "W_i" in different pre-training strategies. a) In Bilingual Expert, one reconstruction loss is computed for each token, conditioned on the entire target context provided by the Forward and Backward decoders. b) With Elmo-Style, it is equivalent to training bidirectional translation models. Two reconstruction losses are computed for each token, each only depending on one side of the context. c) With BERT-Style, certain inputs are masked out (colored in grey) and a masked-LM is learned. One reconstruction loss is computed for each masked token.

$\{\overleftarrow{z_1}, \ldots, \overleftarrow{z_{m_t}}\}$ are computed as:

$$\begin{aligned} c_1, \ldots, c_{m_s} &= \text{Encoder}(x_1, \ldots, x_{m_s}), \\ \overrightarrow{z_1}, \ldots, \overrightarrow{z_{m_t}} &= \overrightarrow{\text{Decoder}}(y_1, \ldots, y_{m_t}, c_1, \ldots, c_{m_s}), \\ \overleftarrow{z_1}, \ldots, \overleftarrow{z_{m_t}} &= \overleftarrow{\text{Decoder}}(y_1, \ldots, y_{m_t}, c_1, \ldots, c_{m_s}). \end{aligned}$$

Both encoder and decoders use Transformer (Vaswani et al., 2017) as their backbone for its better performances in machine translation tasks.

After obtaining these representations, the model is trained with the token reconstruction cross-entropy loss for each target token with contextual information from both sides:

$$\log p(y_j \mid y_{i \neq j}, x) = \text{softmax}(\text{ff}([\overrightarrow{z_j}_{-1}; \overleftarrow{z_j}_{+1}])). \tag{2}$$

Here "ff" denotes a feed-forward layer. Note that we cannot use representations that capture y_j, therefore, we use the forward representation of the previous token $\overrightarrow{z_j}_{-1}$ and the backward representation of the next token $\overleftarrow{z_j}_{+1}$.

SOURCE In the Bilingual Expert model, each target token is predicted given all target tokens on both sides. However, this training scheme makes too much information visible to the model, such that the model could predict the target word even without seeing the source sentence.

For example, we can easily infer that the missing word in "He loves playing _____ and his favorite basketball player is Michael Jordan" is "basketball". In another words, too much visible information on the target side provides an inductive bias that pushes the model towards learning a bi-directional language model instead of a translation-like model, by omitting the information on the source sentence.

We want to force our model to exploit the relationship between the source tokens and target tokens. Thus, we no longer make the words on both sides visible to our model at the same time. Instead, we separate the two directions, so that the model must predict each target word depending only on the source sentence and target words on one side. More specifically, we compute two losses, ℓ_1 and ℓ_2. The cross-entropy loss ℓ_1 is derived by predicting the target word y_j based on the source sentence $\{x_0, \ldots, x_{m_s}\}$ and left-side target words $\{y_0, \ldots, y_{j-1}\}$. Another cross-entropy loss ℓ_2 is derived by predicting y_j based on the source sentence and right-side target words $\{y_{j+1}, \ldots, y_{m_t}\}$. This training scheme corresponds to the strategy used in ELMo (Peters et al., 2018), but the difference is that here we condition on additional source information, hence the name SOURce-Conditional Elmo-style (SOURCE) model.

Another method to force the model to attend more to source is using BERT (Devlin et al., 2018), which masks several words and try to predict those words at once. Inspired by the work of Cross-lingual BERT (Lample and Conneau, 2019), we choose to use the structure as shown in Figure 2. It can reduce the information seen by the decoder and force it to condition more on the source sentence. Due to limitations on time and computing resources, we did not manage to produce successful results using BERT. This would be an interesting and promising direction to explore in future work.

Dataset	Parallel	QE		
		train	dev	test
En-De-SMT	32.8M	26299	1000	1926
En-De-NMT		13442	1000	1023
En-Ru	8.0M	15089	1000	1023

Table 1: Statistics of the parallel data and QE data.

From empirical results of SOURCE, we find that although the prediction accuracy on MT parallel data decreases, the final performance on QE increases significantly. This shows that decreasing the visible information makes token-prediction more difficult, and forces the model to learn more useful structures from the data, which in turn becomes features of higher quality for the QE task.

2.3 Model Ensemble

We perform model ensembling by stacking, which means we use the prediction results of different models on the development set as new features, and train a simple regression model to predict the actual development set labels. Finally, the regression model is applied on the predictions of different models on test set. We use ridge regression here as the regression model. We also use grid search and cross-validation to select the regularization weight for ridge regression.

We train both the pre-trained MT module and the QE scorer module with different hyper-parameters to produce different models for ensembling. For the pre-trained MT module, toggled hyper-parameters include number of layers, number of self-attention heads, learning rate, label-smoothing rate, warm-up steps, and dropout rates. For the QE scorer module, toggled hyper-parameters include number of layers, hidden size, percentage of augmented data, encoder type (LSTM or Transformer), and dropout rate.

3 Experiments

3.1 Settings

Our system is evaluated on the WMT18/19 QE sentence-level task. The main metric is the Pearson's r correlation score between predicted scores and ground-truth HTER scores. There are other metrics including Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for scoring and the Spearman's ρ rank correlation for ranking. We evaluate our models on datasets in the WMT 18/19 shared task with different translation systems: WMT-18 En-De-SMT, WMT-18/19 En-De-NMT, WMT-19 En-Ru-NMT. For experiments on WMT-19 data, we report results based on official evaluation results.

Data For the parallel data used in pre-training of the MT module, we collect large-scale parallel corpora for En-De and En-Ru from the WMT-18 translation task. Officially pre-processed data[1] are utilized. To make it compatible with QE data, we re-escaped certain punctuation tokens. To reduce the corpus size, we further apply a more strict filtering step by discarding sentence pairs with too many overlapping words in their source and target sentences (> 0.9 for En-De, > 0.5 for En-Ru). Finally, we obtain 32.1M EN-De and 7.8M En-Ru sentence pairs and mix it with the training set of the QE data (using post-edited sentences as target). Our mixing strategy is to mix one copy of QE data for every 1M of parallel data. The statistics of the mixed parallel data and QE data are summarized in Table 1.

Following Wang et al. (2018), we also prepare artificial data via round-trip translation (Junczys-Dowmunt and Grundkiewicz, 2016, 2017). Since the QE data are obtained with two kinds of translation systems: SMT and NMT, we also prepare two kinds of artificial data. For simplicity, we take the back-translated corpus by the Edinburgh's translation system,[2] which contains 3.6M back-translated sentences for En-De and 1.9M for En-Ru. We further train a SMT system with Moses and decode the English sentence back to German. For NMT, we simply take a pre-trained NMT system (also Edinburgh's system[3]) for decoding.

Implementation We implement our system from scratch in Python with TensorFlow (Abadi et al., 2015) and OpenNMT (Klein et al., 2017). Because of limited resources, we manually search for good hyper-parameters by heuristics evaluated on the development set. The training of the MT module takes around 4 to 5 days and the training of the QE module takes a couple of hours with one GPU.

[1] http://data.statmt.org/wmt18/translation-task/preprocessed/. According to the official script, the data is processed with tokenization, cleaning and truecase with standard Moses scripts.

[2] http://data.statmt.org/rsennrich/wmt16_backtranslations/.

[3] http://data.statmt.org/wmt17_systems/.

System	test 2018 En-De-SMT			test 2018 En-De-NMT			
	Pearson $r \uparrow$	MAE$\downarrow$	RMSE$\downarrow$	Pearson $r \uparrow$	MAE$\downarrow$	RMSE$\downarrow$	Spearman $\rho \uparrow$
Alibaba (ensemble)	**0.7397**	0.0937	0.1362	0.5012	0.1131	0.1742	0.6049
JXNU (ensemble)	0.7000	0.0962	0.1382	0.5129	0.1114	0.1749	0.6052
SOURCE (ours, ensemble)	—	—	—	**0.5474**	0.1123	0.1623	**0.6155**
SOURCE (ours)	**0.6970**	0.1009	0.1409	**0.4956**	0.1197	0.1797	—
Bilingual Expert (our impl.)	0.6645	0.1089	0.1488	0.4447	0.1240	0.1791	—
melaniad	0.4877	0.1316	0.1675	0.4198	0.1359	0.1770	—
cscarton	0.4872	0.1321	0.1714	0.3808	0.1297	0.1785	—

Table 2: Evaluation results on the test sets of WMT-18 En-De-SMT and WMT-18 NMT. The two leading teams only provide ensemble results on 2018 test data. We re-implement Alibaba's single model (Bilingual Expert) and achieved similar results on 2017 data as reported in their paper. We test that Bilingual Expert on 2018 data to make a fair comparison for single model. With limited computational resource, we only run the ensemble for En-De-NMT, because in WMT2019 they only requires NMT submission.

System	En-De-NMT		En-Ru-NMT	
	Pearson $r \uparrow$	Spearman $\rho \uparrow$	Pearson $r \uparrow$	Spearman $\rho \uparrow$
UNBABEL	0.5718	0.6221	0.5923	0.5388
SOURCE (ours)	0.5474	0.5947	0.4575	0.4039
NJU	0.5433	0.5694	–	–
ETRI	0.5260	0.5745	0.5327	0.5222

Table 3: Evaluation results on the WMT-19 QE sentence-level shared task. Here we only show the top four teams.

3.2 Results

WMT-18 En-De-SMT Results are shown on the left side of Table 2. We can see that our SOURCE model significantly outperforms the state-of-the-art single model from the previous year (Bilingual Expert) and is comparable to the ensemble model from JXNU.

WMT-18 En-De-NMT We evaluate our model through CodaLab, which is recommended by the host. Results are shown on the right side of Table 2. The results are similar to the SMT ones, our single SOURCE model can obtain results comparable to the best ensemble systems. It is worth mentioning that our ensemble model significantly outperforms the best system from the previous year on both scoring (Pearson r) and ranking (Spearman ρ) subtasks.

WMT-19 En-De-NMT and En-Ru-NMT The official result from WMT-19 is shown in Table 3. Our system achieves the second place on En-De and the third place on En-Ru. It is worth mentioning that due to the limitation of computational resource, we train far fewer models for En-Ru than En-De, so it is reasonable that our system performs much better on the En-De dataset.

4 Conclusion and Discussion

Empirical results indicate that decreasing the visible information makes token-prediction more difficult, and forces the model to learn more useful structures from the data, which in turn becomes features of higher quality for the QE task. The experimental results on WMT-18 shows the effectiveness of our SOURCE model as well as our stacking ensemble strategy. According to the official evaluation results on WMT-19 dataset, our ensemble SOURCE modela chieves the second place on En-De dataset and the third place on En-Ru dataset.

We will explore the BERT-style structure to better condition on source sentences in the future.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry

Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 shared task on automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 710–725.

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*.

Kai Fan, Bo Li, Fengming Zhou, and Jiayi Wang. 2018. "bilingual expert" can find translation errors. *CoRR*, abs/1807.09433.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation*, pages 751–758, Berlin, Germany.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. An exploration of neural sequence-to-sequence architectures for automatic post-editing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 120–129, Taipei, Taiwan.

Hyun Kim and Jong-Hyeok Lee. 2016. A recurrent neural networks approach for estimating the quality of machine translation output. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 494–498, San Diego, California.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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.

Jiayi Wang, Kai Fan, Bo Li, Fengming Zhou, Boxing Chen, Yangbin Shi, and Luo Si. 2018. Alibaba submission for WMT18 quality estimation task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 809–815, Belgium, Brussels.