Improving Quality Estimation of Machine Translation by Using Pre-trained Language Representation

Guoyi Miao[1], Hui Di[2], Jinan Xu[1](✉), Zhongcheng Yang[3], Yufeng Chen[1], and Kazushige Ouchi[2]

[1] School of Computer and Information Technology, Beijing Jiaotong University,
Beijing, China
{gymiao, jaxu, chenyf}@bjtu.edu.cn
[2] Toshiba (China) Co., Ltd., Beijing, China
dihui@toshiba.com.cn, kazushige.ouchi@toshiba.co.jp
[3] Qihoo 360 Technology Co., Ltd., Beijing, China
yangzhongcheng@360.cn

Abstract. Translation quality estimation (QE) has been attracting increasing attention due to its potential to reduce post-editing human effort. However, QE still suffers heavily from the problem that the quality annotation data remain expensive and small. In this paper, we focus on overcoming the limitation of QE data and explore to utilize the high level latent features learned by the pre-trained language models to reduce the model's dependence on QE data and improve QE performance. Specifically, we propose two strategies to integrate the pre-trained language features into QE model: (1) a mixed integration model, where the pre-trained language features are fed into the QE mode combined with other features; and (2) a constrained integration model, where a constraint mechanism is used to adjust the reporting bias of our first integration model and enhance the robustness of the QE model. Experimental results on WMT17 QE task demonstrate the effectiveness of our approaches.

Keywords: Quality estimation · Machine translation · Pre-trained language model

1 Introduction

Neural Machine Translation (NMT) has become the state-of-the-art approach to machine translation in the recent years [1, 2]. However, the translation results of NMT are still not perfect, due to some big challenges such as the interpretability problem and the low-resource translation issue. To address this problem, human post-edits by applying insertion, deletion, and replacement operations are required on the translation outputs. Thus machine translation QE, which estimates the quality of translation output without reference at various granularity (sentence/word) levels, can play a crucial role for reducing human effort of post-editing.

Most studies treat QE as a supervised regression/classification task and train the QE model with quality-annotated parallel corpora, called QE data. Some of the previous

S. Huang and K. Knight (Eds.): CCMT 2019, CCIS 1104, pp. 11–22, 2019.
https://doi.org/10.1007/978-981-15-1721-1_2

researches [3–5] employ useful QE features based on feature engineering work to improve QE. However, these manual features are usually expensively available. To solve this problem, some neural networks based models have been applied to QE task [6–9]. Among them, the recent bilingual expert model [9], which uses a bidirectional transformer [2] to construct their language model, achieves the state-of-the-art performance on most public available datasets of WMT17/WMT18 QE task.

Although the bilingual expert model performs well in extracting high level joint latent features, it still can't fully learn enough rich language features due to its single and solidified model architecture. On the other hand, recently some promising pre-trained language models have drawn much attention, such as ELMo [10], OpenAI GPT [11], BERT [12] and XLNet [22]. These models adopting diverse model architecture, first pretrain neural networks on large-scale unlabeled text corpora to learn rich language features, and then finetune the models on downstream tasks.

Inspired by these factors, we view the pre-trained language features as a useful supplement to low resource QE data and investigate the strategies of making full use of these features. Specifically, two strategies are proposed in this paper to integrate the pre-trained language representations into QE model:

(1) Mixed integration model: We use the recent bilingual expert model as our basic model and directly feed the pre-trained language features that are combined with the features learned by the bilingual expert model into the quality estimator of the QE model. That is, the pre-trained language representation is concatenated with the language representation of the bilingual expert model as input features for QE.
(2) Constrained integration model: We enhance the above integration model with a constraint mechanism by using bilingual alignment translation knowledge, which aims to adjust the reporting bias [21] of the pre-trained language features and improve the robustness of QE model.

The key contributions of this paper could be summarized as follows:

(1) We propose two simple yet effective strategies to integrate the pre-trained language features into QE models. Moreover, these strategies are of strong commonality and can be seamlessly applied to other QE models.
(2) We conduct extensive experiments on WMT17 sentence level and word level QE task and verify the effectiveness of the proposed method. Furthermore, we comprehensively analyze the effect of various types of pre-trained language models that are used in our models on QE task and conclude the reasons of these significant improvements.

2 Related Work

Our research is related to three topics, including NMT, pre-trained language representation, and QE for machine translation. We discuss these topics in the following.

2.1 Neural Machine Translation

Most Neural Machine Translation models are based on a sequence-to-sequence attentional framework [1, 2, 13–15], which contains an encoder and a decoder with an attention mechanism. Among them, transformer [2] is the dominant NMT model, which still follows the encoder-decoder architecture, but adopts self-attention networks to attend to the context and avoids recurrence completely to maximally parallelize training.

2.2 Pre-trained Language Model

Pre-trained language representations have shown the effectiveness to improve many natural language processing tasks [10–12, 16, 22]. Unlike traditional word type embeddings [17, 18], **ELMo** adopts left-to-right and right-to-left LSTM to train the word representations. Different from ELMo, **GPT** uses a left-to-right architecture, in which the previous tokens are considered in the self-attention layers of the transformer. Unlike GPT, **BERT** adopts a bidirectional transformer, which allows BERT to capture features from left and right context in all layers. Compared with previous models, **XLNet** is essentially order-aware with positional encodings, and it overcomes some limitations of BERT, such as the pretrain-finetune discrepancy.

2.3 Quality Estimation for Machine Translation

In recent years, there are many works using neural models to estimate the quality of machine translation. Kreutzer et al. [6] propose to use the representations of sentences obtained from neural network for word-level QE task. Kim et al. [8] introduce an entirely neural approach, which is based on a bidirectional and bilingual recurrent neural network (RNN) language model. Recently, Fan et al. [9] propose an end-to-end QE framework for automatically evaluating the quality of machine translation. In their model, a bidirectional transformer is used to build their novel conditional language model which is called neural bilingual expert model.

In this paper, we propose two strategies of integrating the pre-trained language features into our QE models, and our models are developed based on the bilingual expert model [9]. But, different from the bilingual expert model, our work focuses on exploring how to effectively use various pre-trained language models with different strategies to improve QE.

3 Method Description

In this section, we will describe our methods in details. We assume that the features learned by the pre-trained language models are highly related to the QE task and they can be viewed as an important supplement to the QE data. Under this assumption, we aim to explore the method of using the pre-trained language representations for QE task. In this research, we propose two strategies to integrate the pre-trained language representations into QE models and introduce two types of models: (1) mixed integration model, and (2) constrained integration model.

3.1 Mixed Integration Model

A pre-trained language model can learn rich and high level latent features on large unsupervised monolingual corpora, thus, a natural idea of exploiting the model comes out, that is, the features learned by the pre-trained language model can be fed into the QE model as input features. For our first method, we take advantage of the pre-trained language model in a simple and straightforward way. Specially, we follow the work [9] and construct our QE framework on the basis of the bilingual expert model. In our framework, we choose a pre-trained language model, such as ELMo, GPT, BERT and XLNet, as the feature extractor of our model respectively.

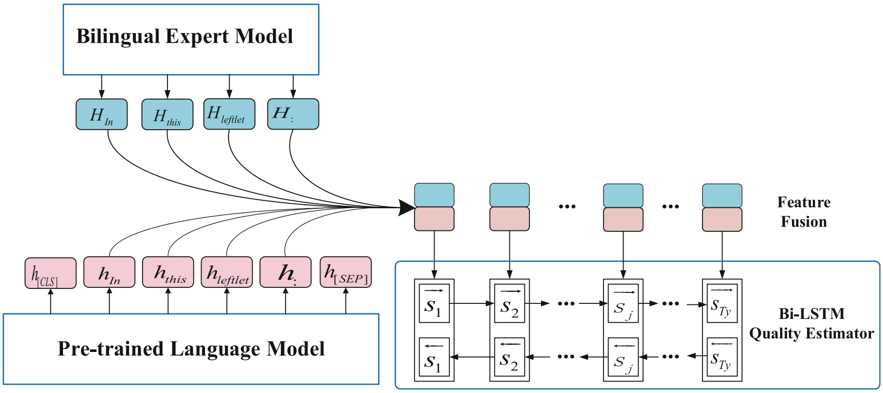

Fig. 1. Illustration of the mixed integration model.

Figure 1 illustrates our mixed integration model. The recent bilingual expert model is used as our baseline model and we directly feed the features learned by the pre-trained language models into the bilingual expert model. Then the feature vector from pre-trained language model is concatenated with the feature vector of the bilingual expert model as input for QE.

After that, the mixed features (from both the pre-trained language model and the bilingual expert language model) will be fed into a bidirectional LSTM quality estimator. For a sentence-level QE task, the hidden layer representation of the last time step is mapped to a real value within interval [0; 1] via a sigmoid function. For a word-level QE task, the hidden layer representation at each time step is mapped to a positive or negative category (‘OK’ or ‘BAD’ tag).

To handle the problem of out-of-vocabulary words, we use WordPiece [19] to segment the input words of the pre-trained language model, like BERT, and each word may be split into several sub-words. For example, the word ORENCIA is split into OR ##EN ##CI ##A, where “##” represents the separator symbol. Since the bilingual expert model does not conduct the segmentation, we add the vectors of several sub-words segmented from an original word, and the sum is used as the hidden layer representation of the original word.

3.2 Constrained Integration Model

Figure 2 illustrates our constrained integration model. The constrained integration model is a modification of the mixed integration model. That is, when predicting quality score, a constraint mechanism is added to adjust the final predicting score, which enhances the robustness of the QE model. Specifically, we extract and introduce bilingual alignment knowledge between source words and target words, which is similar to the information about faithfulness in translation, to adjust the bias of the features learned by the pre-trained language model. The word alignments table, called as $\boldsymbol{A}$, are constructed by using the fast-align tool [20] with both source-to-target and target-to-source directions on bilingual parallel training datasets.

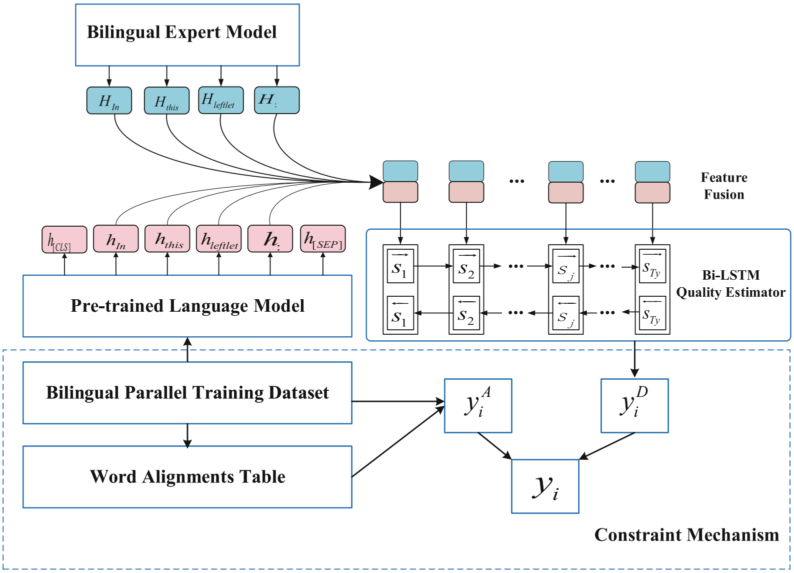

Fig. 2. Illustration of the constrained integration model.

Definition. Given a source sentence $\boldsymbol{X} = \{x_1, x_2, \cdots x_i, \cdots x_N\}$and its corresponding translation sentence $\boldsymbol{T} = \{t_1, t_2, \cdots t_j, \cdots t_K\}$, where $\langle X, T \rangle \in C$, $\boldsymbol{C}$ is the bilingual parallel training dataset, $\boldsymbol{T}$ contains $\boldsymbol{K}$ words and $\boldsymbol{X}$ contains $\boldsymbol{N}$ words. We call word a_i an **alignment word** of word t_j, if $\langle a_i, t_j \rangle \in A$ and $a_i \in X$. Assume all the words in sentence $\boldsymbol{T}$ have a total of $\boldsymbol{N}$ **alignment words**, where $\boldsymbol{N}$ can be statistically analyzed through the word alignments table, and assume that the number of co-occurrences of t_j and its **alignment word** a_i in the bilingual parallel training set $\boldsymbol{C}$ is $\boldsymbol{M}$, t_j appears W times in $\boldsymbol{C}$. Then we define both the sentence level alignment score and word level alignment score as y_i^A. The sentence level alignment score between $\boldsymbol{X}$ and $\boldsymbol{T}$ illustrates the alignment rate between source sentence and its target sentence in translation, and it can be represented as:

$$y_i^A = AlignS(X, T) = N/K \tag{1}$$

where we limit that $AlignS(X, T) \leq 1$.

The word level alignment score between word t_j and sentence $\boldsymbol{X}$ indicates their relevance, and it can be calculated by:

$$y_i^A = AlignW(t_j, X) = M/W \tag{2}$$

For our mixed integration QE model on sentence level QE task, the source sentence $\boldsymbol{X}$ and its corresponding translation $\boldsymbol{T}$ will first be fed into the feature extractor, then the learned hidden representations will be transferred to a bidirectional LSTM quality estimator, after that, a quality score, which can be represented as a real value within interval [0; 1], can be calculated through a sigmoid function:

$$y_i^D = sigmoid(h \cdot U + b) \tag{3}$$

where the sigmoid(·) is a standard nonlinear function; $b \in R$ is a bias term; U represents a parameter matrix; y_i^D is the predictive score for translation result $\boldsymbol{T}$ through our mixed integration model.

However, this predictive value may not be accurate because the features learned by pre-trained language model may be biased. To address this issue, we introduce the bilingual alignment score to adjust the bias. Formally, given a source sentence $\boldsymbol{X}$ and its translation $\boldsymbol{T}$, the final quality score of $\boldsymbol{T}$ can be calculated as follows:

$$y_i = \lambda sigmoid(h \cdot U + b) + (1 - \lambda) AlignS(X, T) \tag{4}$$

where λ represents a weight factor that can be automatically trained by the neural network; y_i is the final predictive score of translation result $\boldsymbol{T}$; h represents a weight parameter, and it can be calculated by:

$$h = \tanh(s \cdot W + b) \tag{5}$$

where s indicates the hidden state at the last time step of the LSTM network; W represents a parameter matrix.

The parameters in these above steps can be optimized through an end-to-end manner with the following object function:

$$loss = 1/n \sum_{i=1}^{n} \sqrt{(y_i - \hat{y}_i)^2} \tag{6}$$

where y_i is the predicted value of the translation result, and $\hat{y}_i$ is the true value.

Notation. For word level QE task, word t_j of the translation $\boldsymbol{T}$ will get a predictive value through Bi-LSTM quality estimator and sigmoid layer, and it will finally be mapped to a positive or negative category (‘OK’ or ‘BAD’ tag). The predictive score of word t_j can be formalized as:

$$y_i = \lambda sigmoid(h \cdot U + b) + (1 - \lambda) AlignW(t_j, X) \tag{7}$$

where h represents the hidden layer representation of word t_j.

4 Experiments

As we have presented above two different strategies to integrate the pre-trained language features into QE models, in the present section we report on a series of experiments on WMT17 QE tasks to test the effectiveness of the proposed strategies.

4.1 Datasets and Evaluation Metrics

We first train the bilingual expert model [9] with large-scale parallel corpus released for the WMT17/WMT18 News Machine Translation Task, which mainly consists of five data sets, including Europarl v7, Europarl v12, Europarl v13, Common Crawl corpus, and Rapid corpus of EU press releases. In addition, the data sets that we use for training the neural bilingual expert model also include parallel corpus released for the WMT17 QE Task, which contains source sentences and their corresponding post-edited translations. It can enable the bilingual expert model to learn more domain knowledge about the QE data. After data cleaning, the final training data contains about 6 M parallel sentence pairs. Then we test the proposed methods on German-to-English (de-en) and English-to-German (en-de) QE tasks. Specifically, we use 0.23 M sentence pairs for training, and 2 K sentence pairs for testing on de-en QE task. For en-de QE task, we use 0.25 M sentence pairs for training, and 2 K sentence pairs for testing.

For pre-trained language models, BERT uses Google’s open source pre-trained version multi_cased Base[1]; ELMo uses the pre-trained Original (5.5B) version[2] of the open source framework AllenNLP; GPT uses open source pre-trained model[3] of OpenAI; and XLNet uses open pre-trained model[4] of Carnegie Mellon University.

In this paper we refer to the QE evaluation metrics of WMT. At sentence level, Pearson, MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and Spearman are used as evaluation metrics. And at word level, we use F1-OK, F1-BAD, and F1-Multi to evaluate QE quality.

4.2 Baselines

To illustrate the effectiveness of our work, we compare our methods with the baseline method as follows:

[1] https://github.com/google-research/bert.

[2] https://allennlp.org/elmo.

[3] https://openai.com/blog/better-language-models.

[4] https://github.com/zihangdai/xlnet.

(1) Bi-Expert: this is the current strongest baseline QE model, called bilingual expert model, which adopts a language model based on a bidirectional transformer and achieves the state-of-the-art performance in most public available datasets of WMT 17/WMT18 QE task.

(2) Bi-Expert+ELMo: this is our mixed integration model, where ELMo is combined with the bilingual expert model as a feature extractor for QE.

(3) Bi-Expert+GPT: this is our mixed integration model, where GPT is combined with the bilingual expert model as a feature extractor for QE.

(4) Bi-Expert+BERT: this is our mixed integration model, where BERT is combined with the bilingual expert model as a feature extractor for QE.

(5) Bi-Expert+XLNet: this is our mixed integration model, where XLNet, the current state-of-the-art pre-trained language model, is combined with the bilingual expert model to produce features for QE.

(6) Bi-Expert+ELMo*: this is our constrained integration model, where a constraint mechanism is used to optimize the objective of integrating ELMo into QE model.

(7) Bi-Expert+GPT*: this is our constrained integration model, where a constraint mechanism is used to optimize the objective of integrating GPT into QE model.

(8) Bi-Expert+BERT*: this is our constrained integration model, where a constraint mechanism is used to optimize the objective of integrating BERT into QE model.

(9) Bi-Expert+XLNet*: this is our constrained integration model, where a constraint mechanism is used to optimize the objective of integrating XLNet into QE model.

It should be noted that, for each of the models described above, (2) to (5) are our mixed integration models, and (6) to (9) are our constrained integration models. The main difference between them is the way they are integrated and the pre-trained language features that are integrated.

4.3 Experimental Settings

The main training settings of bilingual expert model are set as the same as that in the work [9]. Specifically, the vocabulary size is set to 80000; the optimizer uses LazyAdam; the word vector size is set to 512; the block number is set to 2. Besides, the quality estimator adopts a bi-LSTM network, where dropout is set to 0.5, batch size is set to 64, and the hidden layer size is set to 128. To improve the quality of the parallel corpora, we filtered the source and target sentence with length ≤ 70 and the length ratio between 1/3 to 3. We applied byte-pair-encoding (BPE) [23] tokenization to reduce the number of unknown tokens on WMT18 News Machine Translation data sets.

4.4 Experimental Results

Tables 1 and 2 show the QE performance measured at sentence level and word level. It can be seen that, every one of the two QE methods we proposed, by using the pre-trained language features, improves the QE performance over all test sets in comparison to the baseline model-bilingual expert QE model.

Comparison with the Baseline Model. The experimental results in Table 1 indicate that each of the proposed models, whether our mixed integration model or our constrained integration model, can significantly improve the baseline model (bilingual expert model) on sentence level QE task, taking the evaluation metrics Pearson, MAE, RMSE, and Spearman into consideration. Specifically, our best mixed integration model Bi-Expert+XLNet can outperform the baseline model by 0.0154 points in term of Pearson's value, and our best constrained integration model Bi-Expert+XLNet* can improve the baseline model by 0.0206 points in term of Pearson's value on WMT17 de-en test data sets of sentence level QE task. Furthermore, at word level, the experimental results in Table 2 can also show the effectiveness of our two proposed methods on QE task. The above experimental results fully verify that the pre-trained language features are effective for the QE task.

Table 1. Comparison with the current strong baseline model (bilingual expert model, called as Bi-Expert) on **WMT17 de-en** test dataset of sentence level QE task. Row 2 to row 5 represent our mixed integration models, and row 6 to row 9 represent our constrained integration models.

# Models	Pearson's ↑	RMSE ↓	MAE ↓	Spearman ↑
1 Bi-Expert	0.6608	0.1577	0.1112	0.6355
2 Bi-Expert+ELMo	0.6643	0.1553	0.1110	0.6384
3 Bi-Expert+GPT	0.6661	0.1516	0.1092	0.6372
4 Bi-Expert+BERT	0.6747	0.1558	**0.0959**	0.6523
5 Bi-Expert+XLNet	**0.6762**	**0.1513**	0.0964	**0.6545**
6 Bi-Expert+ELMo*	0.6657	0.1542	0.1108	0.6376
7 Bi-Expert+GPT*	0.6695	0.1525	0.1041	0.6432
8 Bi-Expert+BERT*	0.6749	**0.1503**	0.0937	0.6539
9 Bi-Expert+XLNet*	**0.6814**	0.1524	**0.0923**	**0.6558**

Table 2. Comparison with the current strong baseline model (bilingual expert model, called as Bi-Expert) on **WMT17 de-en** test dataset of word level QE task.

# Models	F1-BAD	F1-OK	F1-Multi
1 Bi-Expert	0.4586	0.9363	0.4294
2 Bi-Expert+ELMo	0.5185	0.9438	0.4893
3 Bi-Expert+GPT	0.5179	0.9389	0.4888
4 Bi-Expert+BERT	0.5239	0.9405	0.4927
5 Bi-Expert+XLNet	**0.5286**	**0.9471**	**0.5006**
6 Bi-Expert+ELMo*	0.5194	0.9469	0.4918
7 Bi-Expert+GPT*	0.5166	0.9395	0.4853
8 Bi-Expert+BERT*	0.5270	0.9447	0.4979
9 Bi-Expert+XLNet*	**0.5352**	**0.9526**	**0.5098**

Comparison of our Two Proposed Methods. Experimental results in Tables 1 and 2 show that our proposed constrained integration method has better performance than the proposed mixed integration method for QE. Empirically, our best constrained integration model Bi-Expert+XLNet* can outperform the best mixed integration model

Bi-Expert+XLNet by about 0.0052 points in term of Pearson's value in Table 1. This phenomenon illustrates that our proposed constrained integration method can effectively optimize and denoise the pre-trained language features.

4.5 Analysis

The Effect of Pre-trained Language Models on QE Task. From the experimental results, we find out that XLNet and BERT improve the performance of QE more than other models do. We think it is due to the following three points: (1) The pre-trained language representations can contribute to the improvement of QE to some extent; (2) The ability of feature extraction of transformer is stronger than that of LSTM; (3) Bidirectional language model can capture more features than unidirectional language model can do.

Table 3. Results of sentence level QE on **WMT17 en-de** test dataset. Row 1 represents the current strong QE baseline model (bilingual expert model). Both row 2 and row 3 denote our proposed simple QE models that use BERT and XLNet as feature extractor respectively. Unlike our previous QE models, the pre-trained language features are the only source of features for QE in this model.

# Models	Pearson's ↑	RMSE ↓	MAE ↓	Spearman ↑
1 Bi-Expert	0.6842	**0.1453**	**0.1027**	0.7089
2 BERT+LSTM+MLP	0.6745	0.1539	0.1046	**0.7102**
3 XLNet+LSTM+MLP	**0.6857**	0.1486	0.1031	0.7054

Why Pre-trained Language Models Can Work? Experimental results on WMT17 sentence level and word level QE tasks show that the pre-trained high level latent language features learned by the pre-trained language model can contribute to the improvement of QE. However, this improvement is likely due to the use of a strong baseline system - bilingual expert model, since all of the proposed models are developed based on the bilingual expert model. To verify this assumption is not valid, we construct a simple additional QE model, which only consists of a pre-trained language mode, a LSTM and a Multilayer Perceptron (MLP) neural network, without using the bilingual expert model. The high-level joint features learned by a pre-trained language model are fed into a LSTM and a Multilayer Perceptron (MLP) neural network, and end up with a sigmoid function for estimating quality scores/categories. The experimental results on WMT17 en-de sentence level QE task are shown in Table 3. It is interesting that we find out the performance achieved by the two additional QE models (row 2 and row 3) is close to the performance achieved by the strong baseline model. We believe the reason for the improvement of QE is due to the strong feature learning ability of the pre-trained model itself. The pre-trained language model has learned a wealth of lexical, syntactic and semantic knowledge based on large corpus, so it can effectively alleviate the problem of feature sparseness of QE task.

5 Conclusion and Future Work

In this paper, we attempt to explore how to effectively improve QE with pre-trained language features learned by the pre-trained language models, and propose two strategies to integrate the pre-trained language features into QE models: (1) a mixed integration model, and (2) a constrained integration model. The first model uses a mixed method to treat the pre-trained language model as the feature extractor for QE model, and the second model is enhanced based on our first mixed integration model, which adjusts and optimizes the first model by using bilingual alignment knowledge. Experimental results on WMT17 QE task show that our proposed strategies can significantly improve the translation QE quality. In particular, our strategies are of strong commonality and can be seamlessly applied to other QE models.

In the future, we will explore how to apply transfer learning methods to QE task.

Acknowledgements. This work is supported by the National Nature Science Foundation of China (Nos. 61370130, 61976015, 61473294 and 61876198), the Fundamental Research Funds for the Central Universities (2015JBM033), the International Science and Technology Cooperation Program of China under grant No. K11F100010, the Fundamental Research Funds for the Central Universities (No. 2018YJS043), Major Projects of Fundamental Research on Philosophy and Social Sciences of Henan Education Department (2016-JCZD-022), and Toshiba (China) Co., Ltd.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of ICLR 2015 (2015)
2. Vaswani, A., et al.: Attention is all you need. arXiv preprint arXiv:1601.03317 (2017)
3. Felice, M., Specia, L.: Linguistic features for quality estimation. In: Proceedings of the 7th Workshop on Statistical Machine Translation, pp. 96–103. Association for Computational Linguistics (2012)
4. Specia, L., Shah, K., de Souza, J.G., Cohn, T.: QuEst - a translation quality estimation framework. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 79–84. Association for Computational Linguistics (2013)
5. Kozlova, A., Shmatova, M., Frolov, A.: YSDA participation in the WMT 2016 quality estimation shared task. In: Proceedings of the 1st Conference on Machine Translation, pp. 793–799. Association for Computational Linguistics (2016)
6. Kreutzer, J., Schamoni, S., Riezler, S.: QUality estimation from ScraTCH (QUETCH): deep learning for word-level translation quality estimation. In: Proceedings of the 10th Workshop on Statistical Machine Translation, pp. 316–322. Association for Computational Linguistics (2015)
7. Martins, A.F.T., Astudillo, R., Hokamp, C., Kepler, F.: Unbabel's participation in the WMT16 wordlevel translation quality estimation shared task. In: Proceedings of the 1st Conference on Machine Translation, pp. 806–811. Association for Computational Linguistics (2016)

8. Kim, H., Jung, H.-Y., Kwon, H., Lee, J.-H., Na, S.-H.: Predictor-estimator: neural quality estimation based on target word prediction for machine translation. ACM Trans. Asian Low-Resour. Lang. Inf. Process. (TALLIP) **17**(1), 3 (2017)
9. Fan, K., Wang, J., Li, B., et al.: “Bilingual Expert” can find translation errors. In: National Conference on Artificial Intelligence (2019)
10. Peters, M.E., Neumann, M., Iyyer, M., et al.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
11. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding with unsupervised learning. Technical report, OpenAI (2018)
12. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
13. Wu, Y., et al.: Google’s neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
14. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. arXiv preprint arXiv:1601.03317 (2017)
15. Luong, M.-T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of EMNLP 2015, pp. 1412–1421 (2015)
16. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: Advances in Neural Information Processing Systems, pp. 3079–3087 (2015)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS (2013)
18. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP (2014)
19. Wu, Y., Schuster, M., Chen, Z., et al.: Google’s neural machine translation system: bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
20. Dyer, C., Chahuneau, V., Smith, N.A.: A simple, fast, and effective reparameterization of IBM model 2. In: Proceedings of NAACL 2013 (2013)
21. Gordon, J., Van Durme, B.: Reporting bias and knowledge acquisition. In: Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, pp. 25–30. ACM (2013)
22. Yang, Z., Dai, Z., Yang, Y., et al.: XLNet: generalized autoregressive pretraining for language understanding. arXiv preprint arXiv:1906.08237 (2019)
23. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of ACL 2016, pp. 1715–1725 (2016)