Improved Quality Estimation of Machine Translation with 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) is a task of estimating the quality of translation output from an unknown machine translation (MT) system without reference at various granularity (sentence/word/phrase) levels, and it has been attracting much attention due to the potential to reduce post-editing human effort. However, QE suffers heavily from the fact that the quality annotation data remain expensive and small. In this paper, we focus on the limited QE data problem and seek to find how to utilize the high level latent features learned by the pre-trained language models for improving QE. Specifically, we explore three strategies to integrate the pre-trained language representations into QE models: (1) a mixed integration model, where the pre-trained language features are mixed with other features for QE; (2) a direct integration model, which regards the pre-trained language model as the only feature extracting component of the entire QE model; and (3) a constrained integration model, where a constraint mechanism is added to optimize the quality prediction based on the direct integration model. Experiments and analysis presented in this paper demonstrate the effectiveness of our approaches on QE task.

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

1 Introduction

Neural Machine Translation (NMT) has achieved impressive progress in the recent years with the introduction of efficient architectures, ranging from recurrent [1] to self-attentional networks [2]. However, NMT still faces some big challenges, such as the limited vocabulary size and low-resource translation issues, and thus its outputs are still not perfect. To meet the real-world applications, the translation outputs always require a lot of human post-edits by applying insertion, deletion, and replacement operations. Quality estimation, which estimates the translation quality by predicting the global

J. Tang et al. (Eds.): NLPCC 2019, LNAI 11838, pp. 406–417, 2019.
https://doi.org/10.1007/978-3-030-32233-5_32

sentence quality score or the fine-grained word “OK/BAD” tags, can play a key role for guiding manual correction and 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 researches [3–5] are based on feature engineering work that discovers or designs useful QE features, such as linguistic features, baseline features and pseudo-reference features and feeds them into an estimator for estimating translation quality scores/categories. However, these manual features are usually expensively available. In order to reduce the burden of manual feature engineering, some methods based on neural models have been applied to QE [6–9]. Among them, the classical predictor-estimator model [8] is a recurrent neural network (RNN) architecture that uses a bidirectional and bilingual RNN language model to capture features for the estimator. Different from predictor-estimator model, the recent bilingual expert model [9], which adopts a bidirectional transformer [2] to construct their language model, achieves the state-of-the-art performance in most public available datasets of WMT 2017/2018 QE task.

Although bilingual expert model proposes an effective strategy to enable it to extract high level joint latent features, the limitation of this model is that it cannot flexibly learn enough features from large-scale unsupervised corpus due to its fixed network framework. On the other hand, recently some promising pre-trained language models, such as ELMo [10], OpenAI GPT [11] and BERT [12], which are trained on large unsupervised monolingual corpora and can extract latent rich features, have been applied to many downstream natural language processing (NLP) tasks due to their attractive performance of feature extraction. Apparently, a natural idea is that we use pre-trained language models to obtain features that are useful for QE task.

In this paper, we view the pre-trained language feature as a useful supplement of the existing QE model and investigate how to make full use of these features to improve QE. Specifically, three 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 baseline model and feed the pre-trained language features into the bilingual expert model in a mixed way. That is, the feature representation of pre-trained language model is concatenated with the feature representation of the bilingual expert model as input for QE.
(2) Direct Integration Model: This is a simple yet useful QE model that consists of a pre-trained language representation module, a LSTM layer and a multilayer perceptron (MLP) neural network, where the pre-trained language model is considered as the only feature extracting component of the entire QE model.
(3) Constrained Integration Model: We develop the above direct integration model with a constraint mechanism, which can adjust and optimize the quality prediction of the translation result.

The proposed models assume that the pre-trained language features are highly related to the QE task and they can be regarded as a useful supplement of the exiting QE models. Under this assumption, we believe that the pre-trained language representations can effectively improve QE models.

The key contributions of this paper are listed as follows:

(1) We propose three simple yet effective strategies to integrate the pre-trained language representations into QE models. Moreover, these strategies are of high generality and can be easily applied to other existing QE models.
(2) We conduct extensive experiments on WMT17 QE task and verify the effectiveness of the proposed method.

2 Related Work

Our research is partly built upon a bidirectional transformer-based end-to-end QE model [9], but is also related to Neural Machine Translation (NMT) and pre-trained language representation. We discuss these topics in the following.

2.1 Neural Machine Translation

Generally, most Neural Machine Translation models are based on a sequence-to-sequence attentional framework [1, 2, 13, 14], which contains an encoder and a decoder with an attention mechanism. The encoder, with the help of attention mechanism, summarizes the source sentence into a low-dimensional context vector from which the decoder generates the target sentence word by word. Here are two types of popular NMT models:

RNMT. The RNN-based NMT models [1, 15] are referred as RNMT models, which consists of an encoder RNN and a decoder RNN, interacting via an attention mechanism.

Transformer. Currently, Transformer [2] is the dominant NMT model. Similar to RNMT, the transformer model still follows the encoder-decoder architecture. But unlike RNMT, Transformer makes pervasive use of self-attention networks to attend to the context and avoids recurrence completely to maximally parallelize training.

2.2 Pre-trained Language Representation Models

Pre-trained language representations have shown the effectiveness to improve many natural language processing tasks [10–12, 16]. Recently, some work has attracted much attention due to their significant effects, such as ELMo, OpenAI GPT and BERT. The work has greatly improved the downstream tasks for applying pre-trained language representations.

ELMo. Different from traditional word type embeddings [17, 18], ELMo uses double-layer left-to-right and right-to-left LSTM to train the word representations with a coupled language model (LM) objective, which allows it to learn rich word representations from larger context.

OpenAI GPT. Unlike ELMo, OpenAI GPT uses a left-to-right architecture, in which the previous tokens are considered in the self-attention layers of the Transformer.

BERT. Compared with GPT, BERT adopts a bidirectional Transformer, which allows it to capture features from left and right context.

Following pre-training methods, we refer to the above three work and attempt to integrate the pre-trained language representations into our translation QE models respectively (see Sect. 3). We also comprehensively analyze the effects of various integration methods (see Sect. 4).

2.3 Quality Estimation for Machine Translation

Most of the conventional studies on QE are extensively based on feature engineering work that captures or designs rich QE features as input for regression/classification modules to estimate translation quality scores/categories [4].

In recent years, there are many works that use neural models to estimate the quality of machine translation output. Kreutzer et al. [6] propose using the representations of sentences obtained from neural network, combined with some manually designed features, as input features for word-level QE task. Kim et al. [8] propose an entirely neural approach, called the predictor-estimator architecture, which is based on a bidirectional and bilingual recurrent neural network (RNN) language model. Inspired by the idea of Transformer, Kai et al. [9] propose an end-to-end QE framework for automatically evaluating the quality of machine translation output. In their framework, a bidirectional transformer is used to construct their novel conditional language model called "neural bilingual expert" model, which is trained on a large parallel corpus to extract the high level joint features between the source language and the translation for the downstream QE tasks. The authors show that their bilingual expert model achieves the state-of-the-art performance in most public available datasets of WMT 2017/2018 QE task.

Following the idea of pre-trained language model, in our mixed integration model, we adopt the bilingual expert model as our baseline model and boost this model with some pre-trained language features learned by ELMo, GPT and BERT.

3 Method Description

In this section, we will introduce our methods in details. The proposed methods assume that the features which are learned by the pre-trained language models are highly related to the QE task and they can be regarded as a useful supplement of the exiting expert models. Under this assumption, we aim to explore the method of using the pre-trained language representations on QE task. In this research, we concentrate on the following three strategies to integrate the pre-trained language representations into QE models.

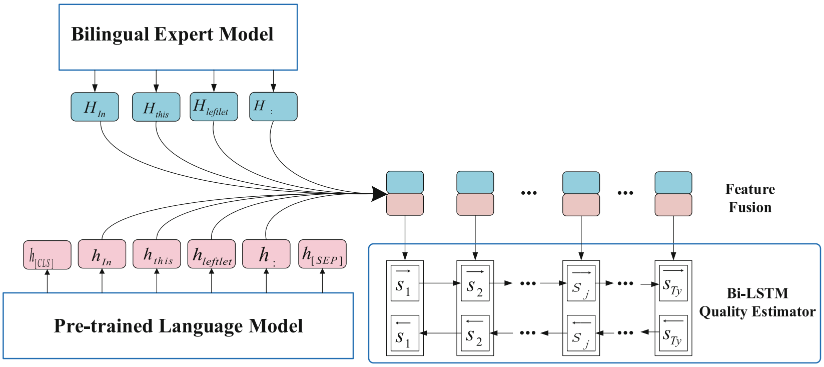

Fig. 1. Illustration of the mixed integration model.

3.1 Mixed Integration Model

For our first method, we follow the work [9] and construct our QE framework based on the bilingual expert QE model, and in our framework, we choose pre-trained language model ELMo, GPT and BERT as the feature extractor respectively. Then the generated features are combined with the features produced by the bilingual expert model as input for QE.

Figure 1 illustrates the mixed integration model. First, we input the translation sentences that to be evaluated into the pre-training language model and the bilingual expert model. The high level joint hidden feature representation h_i learned by pre-trained language model is concatenated with the feature representation H_i learned by the bilingual expert model, which generates the mixed feature representation $[h_i\,;H_i]$. Then the mixed features will be fed into a bidirectional LSTM quality estimator. For a sentence-level QE task, we map the hidden layer representation of the last time step to a real value within interval [0; 1] via a sigmoid function, which can be calculated by:

$$y_i = sigmoid(s^* \cdot U + b) \tag{1}$$

where the sigmoid(·) is a standard nonlinear function; $b \in R$ is a bias term; U represents a parameter matrix; s^* indicates the hidden state at the last time step of the LSTM network; y_i is the predictive score to a machine translation sentence.

Note that, 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).

For sentence level task, the parameters in these above steps can be optimized through an end-to-end manner with the following objective function:

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

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

In addition, to handle the problem of out-of-vocabulary words, we use WordPiece [19] to segment the input words of the pre-trained language model.

3.2 Direct Integration Model

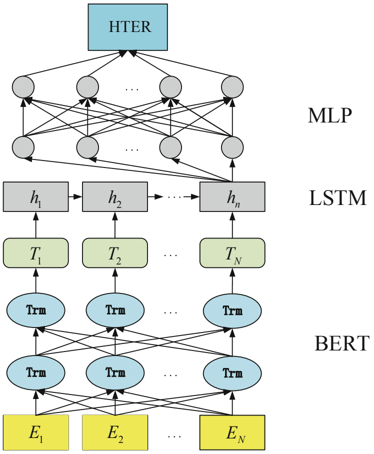

Fig. 2. Illustration of the direct integration model.

The direct integration model is a novel QE architecture based on the pre-trained language model BERT without using the bilingual expert model. Unlike our mixed integration model, in this model, the pre-trained language model is a feature extractor and it is the only source of features for QE model. Moreover, we choose BERT as the feature extractor due to its attractive feature representation ability on sentence level and multilingual learning ability and it can capture the bilingual feature.

As shown in Fig. 2, for sentence level QE task, the source sentences and their corresponding translation (target sentences) are firstly entered into BERT. Then the high-level bilingual joint features learned by BERT are fed into a LSTM network and the hidden layer representation of the last time step is fed into the next Multilayer Perceptron (MLP) neural network. After that, the model ends up with a sigmoid function for estimating quality scores to a translation sentence, and the predictive score y_i^D can be calculated by:

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

where s represents the output of the MLP, and it can be computed as follows:

$$s = \tanh(h^* \cdot W + b) \tag{4}$$

where the $\tanh(\cdot)$ is a standard nonlinear function; h^* indicates the hidden state at the last time step of the LSTM network; W represents a parameter matrix. Note that for word level QE task, h^* is the hidden state corresponding to the current word.

3.3 Constrained Integration Model

The direct integration model that relies on the pre-trained language model may cause it to learn some biased features, and it does not adequately consider the alignment knowledge of parallel sentence pairs. Thus, our constrained integration model is enhanced based on the direct integration model. That is, when predicting quality scores, a constraint mechanism that uses alignment knowledge is added to adjust the final predictive score. We construct the word alignments table ***A*** by using the fast-align tool [20] with both source-to-target and target-to-source directions on bilingual parallel training datasets.

Definition. Given a source sentence $\boldsymbol{X} = \{x_1, x_2, \cdots x_i, \cdots x_N\}$ and its corresponding translation sentence $\boldsymbol{T} = \left\{t_1, t_2, \cdots t_j, \cdots t_K\right\}$, where $\langle X, T\rangle \in C$, $\boldsymbol{C}$ is a bilingual parallel training dataset, $\boldsymbol{T}$ contains $\boldsymbol{K}$ words and $\boldsymbol{X}$ contains $\boldsymbol{N}$ words. We call word x^* the alignment word of the word t_j, if $\langle x^*, t_j\rangle \in A$ and $x^* \in X$. Assume that all the words in sentence $\boldsymbol{T}$ have a total of $\boldsymbol{N}$ alignment words, the number of co-occurrences of t_j and its alignment word x^* in the dataset $\boldsymbol{C}$ 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:

$$y_i^A = AlignSen(X, T) = N/K \tag{5}$$

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

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

$$y_i^A = AlignWord(t_j, X) = M/W \tag{6}$$

In our constrained integration model, we develop the direct integration model by integrating the above sentence level alignment knowledge or word level alignment knowledge. Specifically, we optimize the quality prediction score of the direct integration model by using the bilingual alignment score with a weight factor λ.

Formally, for word level QE task, given a source sentence $\boldsymbol{X}$ and its translation $\boldsymbol{T}$, the final translation quality score of $\boldsymbol{T}$ can be calculated as follows:

$$y_i = \lambda y_i^D + (1-\lambda) y_i^A = \lambda sigmoid(s \cdot U + b) + (1-\lambda) AlignSen(X, T) \tag{7}$$

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.

In addition, for word level QE task, word t_j of the translation $\boldsymbol{T}$ will get a predictive score before it is finally mapped to a positive or negative category (‘OK’ or ‘BAD’ tag). The predictive score for word t_j can be formalized as:

$$y_i = \lambda y_i^D + (1-\lambda) y_i^A = \lambda sigmoid(s \cdot U + b) + (1-\lambda) AlignWord(t_j, X) \tag{8}$$

4 Experiments

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

4.1 Experimental Settings

Our experimental data are divided into two parts, the parallel corpus to train the bilingual expert model, and the QE data based on the WMT17 QE task. The former is mainly obtained from the open news datasets of the WMT17 and WMT18 MT evaluation tasks, including five data sets: Europarl v7, Europarl v12, Europarl v13, Common Crawl corpus, and Rapid corpus of EU press releases. After data cleaning, the final training data totaled about 6M parallel sentence pairs. Then we test the proposed methods on German-to-English (de-en) QE task. Specifically, we use 0.23M sentence pairs for training, and 2K sentence pairs for testing on de-en QE task. It is noted that the main training settings of bilingual expert model are the same as 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, etc. Besides, the quality estimator adopts a bi-LSTM network, where dropout is set to 0.5, batch_size is set to 64, and hidden layer size is set to 128.

In addition, BERT uses Google's open source pre-trained version multi_cased Base[1]; ElMo uses the pre-trained Original[2] (5.5B) version of the open source framework AllenNLP; and GPT uses open source pre-training model[3] of OpenAI.

In this paper we refer to the WMT standard. At the sentence level, Pearson, MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and Spearman are used as evaluating merits. At word level, we use F1-OK, F1-BAD, and F1-Multi to evaluate.

We compare our method with other relevant methods as follows:

(1) Bi-Expert: this is the current strongest baseline 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 2017/2018 QE task.
(2) Bi-Expert+ElMo: this is our mixed integration model, where ElMo is combined with bilingual expert model as a feature extractor for QE.
(3) Bi-Expert+GPT: this is our mixed integration model, where GPT is combined with bilingual expert model as a feature extractor for QE.
(4) Bi-Expert+BERT: this is our mixed integration model, where BERT is combined with bilingual expert model as a feature extractor for QE.
(5) BERT+LSTM+MLP: this is our direct integration model, where BERT is the only feature extracting component of the entire QE model.

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

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

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

(6) BERT+LSTM+MLP*: this is our constrained integration model, where a constraint mechanism is added to optimize the quality prediction.

4.2 Experimental Results

Tables 1 and 2 display the QE performance measured at sentence level and word level. Clearly, our proposed models achieve great improvement on WMT17 sentence level and word level QE task in comparison to the strong baseline system.

Table 1. Comparison with the current strong baseline model (bilingual expert model, called as Bi-Expert) on WMT2017 de-en test dataset of sentence level QE task. Row 2 to row 4 represent our mixed integration models, row 5 represents our direct integration model and row 6 represents our constrained integration model.

#	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	BERT+LSTM+MLP	0.7206	0.1399	**0.0835**	0.6841
6	BERT+LSTM+MLP*	**0.7345**	**0.1384**	0.0936	**0.6893**

Table 2. Comparison with the current strong baseline model (bilingual expert model, called as Bi-Expert) on WMT2017 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	BERT+LSTM+MLP	0.4430	0.9440	0.4182
6	BERT+LSTM+MLP*	0.4627	**0.9456**	0.4375

Comparison with the Baseline System. The results in Table 1 indicate that all our mixed integration models outperform the baseline model (bilingual expert model) taking the evaluation metrics Pearson, MAE, RMSE, and Spearman into consideration. Our best model Bi-Expert+BERT outperforms the baseline model by 0.0139 Pearson's score on WMT2017 de-en test data sets at sentence level. Furthermore, at word level, our best model Bi-Expert+BERT also improves the baseline by 0.0633 F1-Multi points. At sentence level, our best model BERT+LSTM+MLP* can improve the baseline by 0.0737 Pearson's points.

Additionally, the results in Table 1 show that both our direct integration model and the constrained integration model perform well on WMT2017 de-en test data of sentence level QE task, and the constrained integration model can effectively improve the direct integration model by introducing bilingual alignment knowledge.

4.3 Analysis

Table 3. Comparative experiments of our direct integration model on WMT2017 de-en test dataset (sentence level QE task). BERT(target)+LSTM+MLP model is trained with only target monolingual corpus, BERT+LSTM+MLP is trained with bilingual corpus.

#	Models	Pearson's ↑	RMSE ↓	MAE ↓	Spearman ↑
1	BERT(target)+LSTM+MLP	0.6985	0.1552	0.0912	0.6305
2	BERT+LSTM+MLP	**0.7206**	**0.1399**	**0.0835**	**0.6841**

From the experimental results, we may find that BERT improves more than GPT, and GPT improves more than ELMo. We think it is due to the following three points. (1) The ability of feature extraction of transformer is stronger than LSTM. (2) The deeper the network is, the stronger ability of feature extraction it has. (3) Bidirectional language model can capture more information than unidirectional language model.

The results in Table 3 show that in our direct integration model, BERT that is trained with bilingual corpus can contribute better results than that is trained with only target monolingual corpus. We believe that the reason why BERT is effective on sentence-level QE task is that BERT learns the fluency information of sentences through large-scale corpus training. On the other hand, the results in Table 2 indicate that this model is flawed on word-level QE task. We speculate that this is because the pre-trained language model does not learn bilingual translation knowledge.

The experimental results show that our three models achieve great improvement on WMT17 QE task. We believe it is due to the strong representation learning ability of the pre-trained model itself. The pre-trained language model has been pre-trained on large corpus, and the model has learned a wealth of lexical, syntactic and semantic knowledge, 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 utilize the pre-trained language features for improving QE, and explore three strategies to integrate the pre-trained language representations into QE models: (1) a mixed integration model; (2) a direct integration model, and (3) a constrained integration model. The first model uses the pre-trained language model with a mixed method, the second model views the pre-trained language model as the only feature extracting component of the entire QE model, and the third model adjusts and optimizes the second model by using bilingual alignment knowledge. Experimental results on WMT2017 QE task show that our proposed strategies can significantly improve the translation QE quality. Furthermore, our strategies using pre-trained models for QE are of high generality and can be easily applied to other existing QE models.

In the future, we will continue to 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, 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., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N.: Kaiser: 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. Association for Computational Linguistics, pp. 96–103 (2012)
4. Specia, L., Shah, K., de Souza, J.G.C., 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'16 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-Resource Lang. Inform. 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)