NJU's submission to the WMT20 QE Shared Task

Qu Cui and **Xiang Geng** and **Shujian Huang*** and **Jiajun Chen**
National Key Laboratory for Novel Software Technology, Nanjing University
{cuiq,gx}@smail.nju.edu.cn, {huangsj,chenjj}@nju.edu.cn

Abstract

This paper describes our system of the sentence-level and word-level Quality Estimation Shared Task of WMT20. Our system is based on the QE Brain, and we simply enhance it by injecting noise at the target side. And to obtain the deep bi-directional information, we use a masked language model at the target side instead of two single directional decoders. Meanwhile, we try to use the extra QE data from the WMT17 and WMT19 to improve our system's performance. Finally, we ensemble the features or the results from different models to get our best results. Our system finished fifth in the end at sentence-level on both EN-ZH and EN-DE language pairs.

1 Introduction

Quality Estimation (QE) is a task to predict the quality of translations without relying on any references. QE plays a critical role in machine translation to reduce human efforts, such as deciding whether a translation is good enough for post-editing and indicating what edits are needed. This paper describes our system of the Shared Task on Word and Sentence-Level (QE Tasks 2) at WMT20. With the post-edited translations, all the quality scores can be computed automatically by TER-COM (Snover et al., 2006).

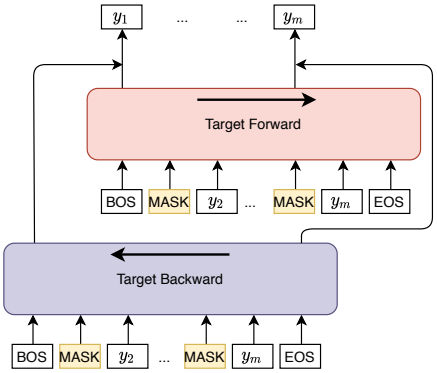
Traditional QE models (Kozlova et al., 2016) use some time-consuming and expensive hand-craft features to represent the translation pairs. With the great success of deep neural networks in natural language processing (NLP), some researches have begun to apply automatic neural features to do QE tasks (Chen et al., 2017; Shah et al., 2016). However, the rare QE data can't fully release the power of deep neural networks. To address this problem, researchers try to transfer bilingual knowledge from parallel data to QE tasks (Fan et al., 2018). These works usually follow a predictor-estimator framework (Kim et al., 2017). This framework first trains the predictor to predict each token of the target sentence given the source and the context of the target sentence on parallel data. Then, the estimator is trained using the features of QE data produced by the predictor.

However, existing predictor-estimator frameworks cannot fully use the information from parallel data because of the discrepancy of data quality between the predictor and the estimator. The predictor is trained on parallel data, which are nearly no errors in translations. While the translations in QE data is generated by a real machine translation system and may have some errors. When the estimator is training on the QE data, the predictor needs to extract the features of translations with some errors, which is quite different from the parallel data. Thus, the predictor can't extract features well.

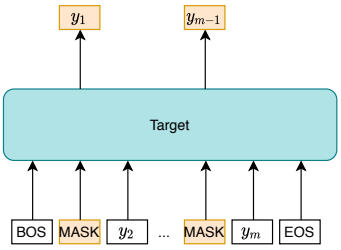
To fix this problem, we present two different approaches in this paper. The first model masks some tokens at the target side but still need to predict every token correctly, and it enhances the ability of the model to deal with translations with errors. And to obtain the deep bi-directional information, we use a masked language model at the target side instead of two single directional decoders. Meanwhile, we try to use the extra QE data, which are from the WMT17 and WMT19 to improve our system's performance. Finally, we ensemble the features or the results from different models to get our best results. Our system finished fifth in the end at sentence-level on both EN-ZH and EN-DE language pairs of the WMT20 QE shared tasks (Specia et al., 2020).

* Corresponding Author.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1004–1009
Online, November 19–20, 2020.



(a) QE Brain mask.



(b) MTLM.

Figure 1: To save space, we do not show the source encoders of these models in the figure. (a) shows the QE Brain mask system, and it simply enhances the original QE Brain system by simply masking tokens at the target side. (b) uses a masked language model at the target side to obtain deep bidirectional information.

2 Methods

As we all know, using different sub-models for ensemble will have better results (Krogh and Vedelsby, 1995). We ensemble different methods in our system, some of them are existing methods, and the others are proposed by us. Next, we will describe these methods.

2.1 Existing Methods

2.1.1 QUETCH

QUETCH (Kreutzer et al., 2015) (QUality Estimation from scraTCH) is a multilayer perceptron model trained without auxiliary parallel data. The embeddings of input passed through one linear layer with tanh activation functions and then one output layer with softmax activation functions, one linear layer with tanh activation functions, one output layer with softmax activation functions. QUETCH only outputs OK/BAD probabilities for each word in the word-level task. Similar to (Martins et al., 2017), we estimate HTER with the fraction of BAD labels for the sentence-level task.

2.1.2 NuQE

NuQE (Martins et al., 2016) (NeUral Quality Estimation) can be seen as a stronger version of QUETCH by using complex neural networks. The architecture of NuQE consists of one embeddings layer, one linear layer, one bi-directional GRU layer, two other linear layers. The input and output of NuQE is the same as QUETCH. We use QUETCH and NuQE as implemented in OpenKiwi (Kepler et al., 2019) [1].

2.1.3 QE Brain

QE Brain (Fan et al., 2018) is based on the predictor-estimator framework. The predictor uses transformer neural networks and will be pretrained on the parallel corpus. The model consists of encoder and bi-directional decoder to encode the source sentence $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ and predict each token in the target sentence $\mathbf{Y} = \{y_1, y_2, \ldots, y_m\}$ with the help of hidden representations of the source sentence, respectively.

When training the Bi-LSTM (Graves and Schmidhuber, 2005), which is used as the estimator, the source sentence and translation are fed into the predictor to extract features. Similar to common predictor-estimator methods, QE Brain uses the hidden state of the final layer in the predictor as model derived features. They also extract the difference between the probability of generating the current token and the most likely token as mismatching features. Finally, the estimator concatenate model derived features and mismatching features to predict the word-level tags $\mathbf{O}$ and sentence-level HTER q.

Our proposed models are based on the QE Brain.

2.2 Proposed Methods

2.2.1 Masked QE Brain

Researches used to transfer bilingual knowledge from parallel data to QE tasks, however, the data distribution between parallel data and QE data is different. The translations in QE data are generated by a real machine translation system, and there will be some errors in these translations. While the translations in parallel data generated by humans, and there are nearly no errors. It means, the predictor trained on parallel data can not perform well when it is feeding with translations with errors because the contexts at the target side are different.

[1] https://unbabel.github.io/OpenKiwi.

Pair	Dataset	Train	Dev	Test
EN-DE	WMT20	7,000	1,000	1,000
	WMT19	13,442	-	-
	WMT17	23,000	-	-
EN-ZH	WMT20	7,000	1,000	1,000

Table 1: The statistics of QE dataset used in our system for the WMT20 QE shared task.

Pair	Train	Dev
EN-DE	23,438,059	2,000
EN-ZH	7,460,939	2,000

Table 2: Parallel Dataset statistics used in our system. We divide parallel data into a training set and development set.

To partially alleviate this problem, we proposed the masked QE Brain, as shown in Figure 1(a).

The motivation for our method is simple. We want to enhance the predicting ability of the model in the wrong contexts. To achieve this goal, when training the predictor on parallel data, we mask some tokens in the translation. And the predictor needs to make the same prediction as they are feeding with the complete pair. The other part is the same as the original version of the QE Brain.

2.2.2 Masked Target Language Model

The QE Brain and Masked QE Brain use a bi-directional decoder at the target side to obtain the information from both sides. However, this architecture is just a shallow concatenation which can not truly get the information from both sides (Devlin et al., 2018).

Thanks to the masked tokens in target sentences of Masked QE Brain, we can easily use a masked language model (Devlin et al., 2018) at the target side instead. We call this model the Masked Target Language Model (MTLM), and the format of the input is just the same as Masked QE Brain, as shown in Figure 1(b). They both input the source sentence $\mathbf{X}$, the masked target sentence $\mathbf{Y}'$. And the MTLM only need to predict the right tokens of these masked ones at the target side while Masked QE Brain needs to predict all the tokens.

3 Experiments

3.1 Dataset

3.1.1 Data statistics

QE Dataset The QE tasks of WMT20 contains both EN-DE language pair and EN-ZH language pair. They both have sentence-level and word-level tasks. Meanwhile, the word-level task contains the prediction for source tokens, target tokens, and target taps. In our paper, we only report word-level results on target tokens. In our work, we also use the EN-DE QE dataset of WMT17 and WMT19 to help train an ensemble model. The statistics of QE datasets are shown in Table 1.

Parallel Dataset For the EN-DE language pair, we use the data officially released by the organizers. And for the EN-ZH language pair, we use the parallel data from the WMT18 EN-ZH translation task. The statistics of parallel datasets are shown in Table 2.

3.1.2 Preprocess

EN-DE We use BPE (Sennrich et al., 2015) to segment both the English and German texts, and the BPE step is set to 30,000. We learn the BPE code jointly but build the two vocabularies separately. The size of EN is 14,112; the size of DE is 23,458.

EN-ZH We also use BPE to segment English texts here, and the setting is the same as those in EN-DE. The final size is 34,466. For Chinese texts, we keep all the sentences in the original QE dataset, and then use jieba[2] to segment other Chinese sentences in the parallel dataset. We choose the top 40,000 tokens of the frequency as the vocabulary.

3.2 Settings

Metrics The metric of sentence-level QE is Pearson's Correlation Coefficient. And the metrics of word-level QE are F1-MULT (the products of both positive and negative examples) and Matthews's Correlation Coefficient.

Hyper-parameters

- NuQE. The hidden size is [400, 200, 100, 50].
- QUETCH. The hidden size is [100, 50].
- QE Brain. The predictor contains one encoder and two decoders of 6 layers with 512 hidden units. The estimator is a Bi-LSTM, and its hidden size is 512.

[2] https://github.com/fxsjy/jieba

Pair	Method	Sent-level Dev	Word-level Dev	
			F1-MULT	MCC
EN-DE	NuQE	30.75	37.63	27.41
	QUETCH	31.27	37.19	27.78
	QE Brain	48.70	34.68	28.74
	QE Brain mask	**53.34**	35.15	30.17
	MTLM	49.77	**39.68**	**33.99**
	f-ensemble	59.91	-	-
	r-ensemble	59.76	-	-
	v-ensemble	-	47.58	42.36
EN-ZH	NuQE	42.49	43.50	33.02
	QUETCH	42.97	31.60	30.83
	QE Brain	58.05	44.07	32.85
	QE Brain mask	58.97	46.55	36.50
	MTLM	**60.89**	**51.31**	**43.33**
	f-ensemble	66.02	-	-
	r-ensemble	62.13	-	-
	v-ensemble	-	51.81	45.33

Table 3: Results of WMT20. f-ensemble means we ensemble different methods by features, r-ensemble means we ensemble different methods by their results and v-ensemble means different methods vote for an ensemble result.

- QE Brain mask. It is all the same as the QE Brain.
- MTLM. The predictor contains one encoder and one decoder of 6 layers with 512 hidden units. And the estimator is the same as the QE Brain.

3.3 Singe Model Results

Table 3 shows the single model results of our system. Different models are using the same parallel data and only using the QE dataset of WMT20.

The NuQE and QUETCH are only trained on the QE dataset, while the other methods are also trained on extra parallel datasets. We can see that the performance of NuQE and QUETCH is far from that of these models that have extra bilingual knowledge.

Compare with the original QE Brain, our two proposed models can have a big improvement.

3.4 Data Ensemble

We train to enhance our system by using other QE datasets, mainly from WMT17 and WMT19. We only try this on the EN-DE language pair. As we can see in Table 4, if we use more QE data, the performance can get a big improvement easily.

3.5 Model Ensemble

We also try to ensemble different methods and finally get the best result. For sentence-level, we try two different ways. First, we use QE Brain, QE Brain mask, and MTLM as a feature extractor. The features from the three models will be combined and then used to predict the hter scores. Second, we simply collect the predictions of different methods on the training set, development set and test set. The training predictions will be feed into a dense layer and used to predict true hter score, development predictions will be used to early stop. Finally, we will use the trained dense layer to deal with the test predictions.

For word-level, we simply use voting to ensemble different models. The results are shown in Table 3.

3.6 Final Results

Table 5 shows our final results of WMT20 on the web pages. Our system does not contain predictions on target gaps on the word-level, so we just combine the results on gaps from NuQE and QUETCH and our results on target tokens to build the final result.

Method	Dataset	Sent-level Dev	Word-level Dev	
			F1-MULT	MCC
QE Brain	WMT20	48.70	34.68	28.74
	ensemble	53.44	39.04	35.04
QE Brain mask	WMT20	53.34	35.15	30.17
	ensemble	54.87	40.05	35.25
MTLM	WMT20	49.77	39.68	33.99
	ensemble	53.38	43.40	36.41

Table 4: Ensemble results of the WMT20 EN-DE language pair, we train the QE systems on the combination of WMT20, WMT19, and WMT17 dataset.

Pair	Sent-level	Word-level
EN-DE	61.81 (5th)	45.11 (6th)
EN-ZH	64.23 (5th)	55.13 (6th)

Table 5: Final results and rank of WMT20 on the web page, the sentence-level metric is Pearson's Correlation Coefficient, and the word-level metric is the Matthews's Correlation Coefficient.

4 Conclusion

This paper describes our system of the WMT20 QE shared task. Our work mainly follows the QE Brain. To bridge the gap between parallel data and QE data, we use a simple way to bring noise into target sentences of parallel data. And to achieve deep bidirectional information, we use a masked language model at the target side. Experiments show that our two-step approaches achieve improvements. Meanwhile, we try to train our models on more QE data with the same language pair and ensemble different methods through different ways to get our final results.

References

Zhiming Chen, Yiming Tan, Chenlin Zhang, Qingyu Xiang, and Mingwen Wang. 2017. Improving machine translation quality estimation with neural network features. In *Proceedings of the Second Conference on Machine Translation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Kai Fan, Jiayi Wang, Bo Li, Fengming Zhou, Boxing Chen, and Luo Si. 2018. "bilingual expert" can find translation errors. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Netw*, 18(5-6).

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. Openkiwi: An open source framework for quality estimation.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*.

Anna Kozlova, Mariya Shmatova, and Anton Frolov. 2016. Ysda participation in the wmt'16 quality estimation shared task. In *Proceedings of the First Conference on Machine Translation*.

Julia Kreutzer, Shigehiko Schamoni, and Stefan Riezler. 2015. QUality estimation from ScraTCH (QUETCH): Deep learning for word-level translation quality estimation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*.

Anders Krogh and Jesper Vedelsby. 1995. Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*.

André F. T. Martins, Ramón Astudillo, Chris Hokamp, and Fabio Kepler. 2016. Unbabel's participation in the WMT16 word-level translation quality estimation shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*.

André Martins, Marcin Junczys-Dowmunt, Fabio Kepler, Ramon Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. 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).

Kashif Shah, Fethi Bougares, Loïc Barrault, and Lucia Specia. 2016. Shef-lium-nn: Sentence level quality estimation with neural network features. In *Proceedings of the First Conference on Machine Translation*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. *Proceedings of Association for Machine Translation in the Americas*.

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