NJUNLP's Submission for CCMT20 Quality Estimation Task

Qu Cui[1], Xiang Geng[1], Shujian Huang[1]*, and Jiajun Chen[1]

National Key Laboratory for Novel Software Technology, Nanjing University
{cuiq,gx}@smail.nju.edu.cn, {huangsj,chenjj}@nju.edu.cn

Abstract. Quality Estimation is a task to predict the quality of translations without relying on any references. QE systems are based on neural features but suffer from the limited size of QE data. The best models nowadays transfer bilingual knowledge from parallel data to QE tasks. However, the distribution between parallel data and QE data is different which may lead to that the value of parallel data can not be used for best. More specifically, there are no errors in parallel translations while there may be more than one error in the translations of QE data. To alleviate this problem, we propose a model which will mask some tokens at the target side on parallel data but still need to predict every target token. And based on this model, we propose a variant model that uses a masked language model at the target side to obtain deep bi-directional information. Besides, we also try different ensemble methods to get better performance of the CCMT20 Quality Estimation Task. Our system finally won second place in the ZH-EN language pair and third place in the EN-ZH language pair.

Keywords: Quality Estimation · Data Distribution · Mask.

1 Introduction

Quality Estimation is a task to predict the quality of translations without relying on any references. It has both a word-level and a sentence-level task, all the quality scores are computed automatically by TERCOM [13].

Researchers first use some hand-craft features to represent the translation pairs and do QE tasks [7]. However, it is time-consuming and expensive. Then, automatic neural features are introduced to QE tasks [1, 12]. A remaining problem is that the data of QE tasks is hard to get, and it limits the performance of QE models. To solve this problem, researchers began to transfer bilingual knowledge from parallel data to QE tasks. They usually follow a predictor-estimator framework [6, 3], which first trains a predictor on parallel data and then trains an estimator on QE data based on the features produced by the predictor.

This structure has achieved great success but it still has some problems. The data distribution between QE data and parallel data is different. The translations

* Corresponding Author.

in QE data are generated by a real machine translation (MT) system and there will inevitably be some noise. However, there are nearly no errors in parallel translations. When training the predictor on parallel data, the model may make the right choice only based on the translation. In this case, there will be problems when doing QE tasks since the real translations in QE data are no longer reliable.

To alleviate this problem, we propose a model that will mask some tokens at the target side but still need to predict every token correctly. Such a way will help the model reduce the dependence on the translations when training on the parallel data, and it can enhance the ability of the model to deal with translations with errors. Moreover, to obtain the deep bi-directional knowledge, we use a masked language model at the target side instead of a concatenation of two single directional decoders which is used in the traditional predictor-estimator framework.

We ensembled the existing methods and our proposed methods and participated in the CCMT20 Quality Estimation task. Our system finally won second place in the ZH-EN language pair and the third place in the EN-ZH language pair. Meanwhile, we also conduct experiments to show how our approach works.

2 Methods

In this section, we are going to show the details of the methods used in our final submitted system. They will be divided into two parts. First, we will list the existing methods and second the proposed methods by us.

2.1 Existing Methods

QUETCH As the name implies, QUETCH [8], QUality Estimation from scraTCH, is trained from scratch with only QE data. The architecture of QUETCH consists of one embeddings layer, one linear layer with the tanh activation function, one output layer with the softmax activation function. We use the fraction of BAD labels as an estimate for HTER score at sentence-level [10].

NuQE NuQE [9], NeUral Quality Estimation, carries QUETCH one step further with complex neural networks. Their model architecture consists of a linear layer, a bi-directional GRU layer, two other linear layers. And NuQE is also trained without auxiliary parallel data.

We use QUETCH and NuQE as implemented in OpenKiwi [5] [1].

QE Brain QE Brain [3] follows the predictor-estimator architecture. When training the predictor on parallel data, they first use an encoder based on transformer [14] to encode the source sentence $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ and then use a bi-directional decoder to predict each token in the target sentence $\mathbf{Y} = \{y_1, y_2, \ldots, y_c\}$ with the help of hidden representations of the source sentence.

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

When training on real QE data, they also use the predictor to predict each token in the translation from real translation systems. And the hidden state of the final layer in the predictor will be used as the word-level features. Meanwhile, the probability difference between the probability of generating the current token and the most likely token, which is called mis-matching feature will also be used as the word-level feature. Finally, they use a Bi-LSTM [4] as the estimator to combine the word-level features to predict the word-level tags $\mathbf{O}$ and sentence-level scores q.

Our proposed models are mainly based on the QE Brain.

2.2 Proposed Methods

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 main difference is that, the translations in QE data are generated by a real machine translation system, and there will be some errors. While the translations in parallel data are 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. To partially alleviate this problem, we proposed our Masked QE Brain.

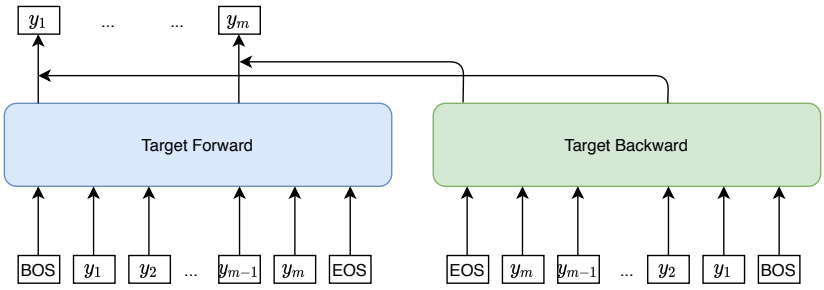
The motivation of our method is very direct, we want to enhance the ability of the model when doing predictions with wrong contexts. In order to achieve this goal, we mask some tokens in the translation when training the predictor on parallel data. And the predictor needs to do the same prediction as they are feeding with the complete pair. The rest parts of our model are the same as those in the original version of the QE Brain.

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 bi-directional information [2]. We use a masked language model [2] at the target side instead to solve this problem and get the deep bi-directional information. We call this model the Masked Target Language Model (MTLM), and the format of the input is just the same as that in the Masked QE Brain. The two models both use the source sentence $\mathbf{X}$, the masked target sentence $\mathbf{Y}'$ as the input. 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.

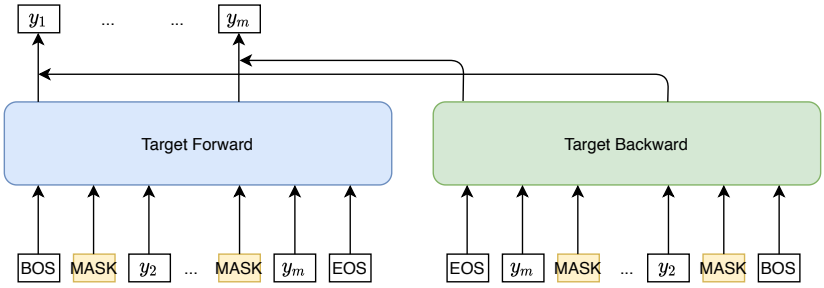
Figure 1 shows the model architecture of the original QE Brain and the two proposed models.

3 Experiments

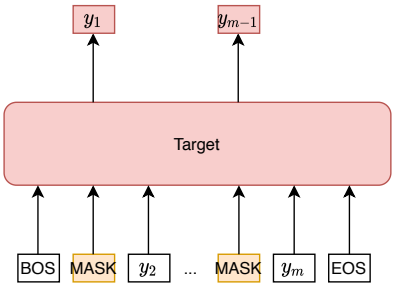
In this section, we will show the details of our experiments, consisting of the dataset, hyper-parameters, performance of single models, and so on.



(a) Original QE Brain.



(b) Masked QE Brain.



(c) MTLM.

Fig. 1. These models have the same source encoder, we do not show it in the figure to save space. (a) shows the original QE Brain, and (b) enhances it by simply masking tokens at the target side. (c) uses a masked language model at the target side to obtain deep bidirectional information.

Direction	Aspect	Train	Dev	Test
EN-ZH	word	10,878	1,128	4,151
	sent	14,789	1,381	4,355
ZH-EN	word	11,017	1,046	4,129
	sent	10,070	1,143	4,211

Table 1. QE Dataset statistics of the CCMT20.

Dataset	Train	Dev
WMT18	7,460,939	2,000
neu2017	1,999,000	1,000
datum2015	999,004	1,000
casia2015	1,049,000	1,000
casict2015	2,035,834	1,000

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

3.1 Dataset

QE Dataset The QE tasks of CCMT2020 have two language directions of both EN-ZH and ZH-EN, and they have two aspects of both word-level and sentence-level. The word-level task contains tags of source tokens, target tokens, and target gaps, and we only have results on target tokens. The statistics of QE datasets are shown in Table 1.

Parallel Dataset We use different parallel datasets in our system. And we do not use all parallel datasets on all of the methods. The statistics of parallel datasets are shown in Table 2.

3.2 Settings

Metrics For the word-level task, the metrics are F1 scores of the products of both positive and negative examples. For the sentence-level task, the metric is the Pearson's Correlation Coefficient.

Hyper-parameters For NuQE and QUETCH, we simply use the software released publicly.

For the original QE Brain and Masked QE Brain, both the encoders and the two decoders have 6 layers of transformers with 512 hidden units. And for

Parallel Dataset	Method	EN-ZH		ZH-EN	
		Sent	Word	Sent	Word
-	NuQE	19.39	**29.22**	23.75	35.56
-	QUETCH	29.08	11.72	25.04	30.24
WMT18	QE Brain	47.26	15.90	52.04	37.68
	Masked QE Brain	47.26	23.38	52.77	35.60
	MTLM	**52.16**	24.67	**55.94**	**43.75**
neu2017	MTLM	45.23	20.07	53.43	40.62
datum2015		44.58	14.49	50.49	41.30
casia2015		44.27	23.43	51.45	42.34
casict2015		39.19	14.74	52.34	42.53
ensemble	sent-neural	56.55	-	57.23	-
ensemble	sent-result	54.18	-	55.18	-
ensemble	word-voting	-	30.25	-	48.28

Table 3. Results of the CCMT20.

MTLM, the model only has one encoder and one decoder which has the same number of layers and units as the original QE Brain. These three models all use Bi-LSTM [4] as the estimator, of which the hidden size is set to 512.

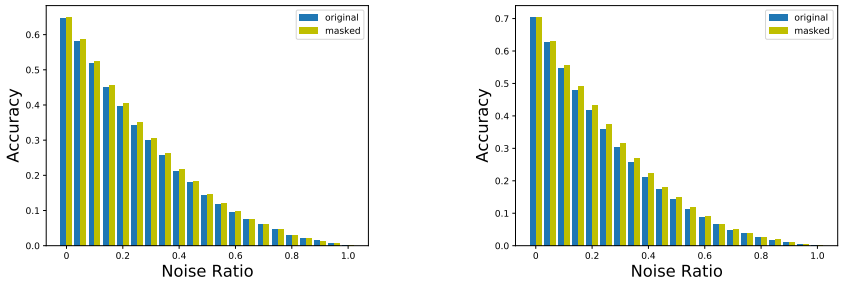
Tokenize We use BPE [11] to tokenize the English dataset and use jieba[2] to tokenize the Chinese dataset. The step of BPE is set to 30,000, and we use all tokens after tokenized. And we use *jieba* to tokenize the Chinese sentences, meanwhile, the tokens in the EN-ZH word-level task will not be tokenized. Finally, we only use the 30,000 most frequent tokens of all Chinese tokens.

3.3 Single Model Results

The results of single models are shown in the Table 3. As we can see, these models without parallel knowledge do not have a good performance except on the word-level task of the EN-ZH direction. When pretraining on the WMT18 parallel dataset, our two proposed methods all perform better than the original QE Brain. And the MTLM has the best performance, due to that it both alleviates the problem of data distribution and obtains deep bi-directional information.

More parallel data will bring better performance. The MTLM's performance pretraining on the data of WMT18 is better than that of the other four datasets, and the size of the WMT18 dataset is almost three times bigger.

[2] https://pypi.org/project/jieba/



(a) EN-ZH. (b) ZH-EN.

Fig. 2. The accuracy of predicting right tokens when a part of target tokens are replaced by random ones.

3.4 Ensemble

We try two different ensemble methods at the sentence-level.

Neural Ensemble The hidden states of QE Brain, Masked QE Brain, MTLM for the same sentence will be gathered, and then an extra linear model will be used to map the hidden states to real HTER values.

Result Ensemble We gather the HTERs of both training datasets and development datasets from all of the models described above. And then we train a linear model that learns to use these HTER values to predict the golden HTER value.

And for the word-level task, we simply use voting to ensemble the results of all models.

The ensemble results are also shown in Table 3. And we can see that the neural ensemble way outperforms the other one at the sentence-level. Our system finally won second place in the ZH-EN language pair and the third place in the EN-ZH language pair.

4 Analysis

In this section, we will discuss the effectiveness of our approach.

SRC	for those toiling far below the surface of the Earth , the proposed system could prove a godsend .
TGT	对于 那些 在 地下 辛苦工作 的 人 来说 ， 这个 新 系统 简直是 神赐 之物 。
Masked TGT	对于 那些 在 地下 [MASK] 的 人 来说 ， 这个 新 系统 简直是 [MASK] [MASK] 。

Table 4. A case of training data in the parallel data.

Table 4 shows an example of our training data in the parallel dataset. As we can see, when training the predictor on the complete target sentence, the model may predict the token '之物' only with the help of token '神赐', because this binary combination is common. However, what we want is that the model can rely less on target sentences. We can easily break the self-dependence by masking tokens in target sentences, and this will enhance the ability of the predictor when feeding with wrong target sentences.

We test the predicting ability of the original QE Brain and our Masked QE Brain when the target sentences are partially replaced by random tokens. The results are shown in Figure 2. And we can see that when there is no noise in target sentences, the two models have a similar performance. As the replacement ratio grows, our Masked QE Brain has a growing advantage in both language directions.

5 Conclusion

This paper describes our systems for CCMT20 Quality Estimate tasks including both word-level and sentence-level.

We follow the predictor-estimator architecture and mainly follow QE Brain. To alleviate the problem that the distribution between parallel data and QE data is different, we proposed the Masked QE Brain. And to achieve the deep bi-directional information, we use a masked language model at the target side and propose our MTLM.

The proposed models perform better than the original version of the QE Brain. At the same time, we use different ensemble methods to achieve our final results for CCMT20. Our system finally won second place in the ZH-EN language pair and the third place in the EN-ZH language pair.

Acknowledgement

The authors would like to thank Yiming Yan for the feedback.

References

1. Chen, Z., Tan, Y., Zhang, C., Xiang, Q., Wang, M.: Improving machine translation quality estimation with neural network features. In: Proceedings of the Second Conference on Machine Translation (2017)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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 (2018)
3. Fan, K., Wang, J., Li, B., Zhou, F., Chen, B., Si, L.: "bilingual expert" can find translation errors. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)

4. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks (2005)
5. Kepler, F., Tr'nous, J., Treviso, M., Vera, M., Martins, A.F.T.: Openkiwi: An open source framework for quality estimation (2019)
6. Kim, H., Lee, J.H., Na, S.H.: Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In: Proceedings of the Second Conference on Machine Translation (2017)
7. Kozlova, A., Shmatova, M., Frolov, A.: Ysda participation in the wmt'16 quality estimation shared task. In: Proceedings of the First Conference on Machine Translation (2016)
8. Kreutzer, J., Schamoni, S., Riezler, S.: QUality estimation from ScraTCH (QUETCH): Deep learning for word-level translation quality estimation. In: Proceedings of the Tenth Workshop on Statistical Machine Translation (Sep 2015)
9. Martins, A.F.T., Astudillo, R., Hokamp, C., Kepler, F.: 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 (2016)
10. Martins, A., Junczys-Dowmunt, M., Kepler, F., Astudillo, R., Hokamp, C., Grundkiewicz, R.: Pushing the limits of translation quality estimation. Transactions of the Association for Computational Linguistics (2017)
11. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (2015)
12. Shah, K., Bougares, F., Barrault, L., Specia, L.: Shef-lium-nn: Sentence level quality estimation with neural network features. In: Proceedings of the First Conference on Machine Translation (2016)
13. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of Association for Machine Translation in the Americas (2006)
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems 30 (2017)