LETTER

A Unified Neural Network for Quality Estimation of Machine Translation*

Maoxi LI[†a)], ***Member***, **Qingyu XIANG**[†b)], **Zhiming CHEN**[†], ***and*** **Mingwen WANG**[†], ***Nonmembers***

SUMMARY The-state-of-the-art neural quality estimation (QE) of machine translation model consists of two sub-networks that are tuned separately, a bidirectional recurrent neural network (RNN) encoder-decoder trained for neural machine translation, called the predictor, and an RNN trained for sentence-level QE tasks, called the estimator. We propose to combine the two sub-networks into a whole neural network, called the unified neural network. When training, the bidirectional RNN encoder-decoder are initialized and pre-trained with the bilingual parallel corpus, and then, the networks are trained jointly to minimize the mean absolute error over the QE training samples. Compared with the predictor and estimator approach, the use of a unified neural network helps to train the parameters of the neural networks that are more suitable for the QE task. Experimental results on the benchmark data set of the WMT17 sentence-level QE shared task show that the proposed unified neural network approach consistently outperforms the predictor and estimator approach and significantly outperforms the other baseline QE approaches.

key words: *natural language processing, machine translation, neural quality estimation, recurrent neural network (RNN), bidirectional RNN encoder-decoder with attention mechanism*

1. Introduction

Quality estimation (QE) of machine translation estimates the quality of translation outputs without the use of human references. It plays an increasingly important role in the post-editing of machine translations and computer-aided translations.

The traditional methods formalize the problem of QE using supervised regression/classification models. One of the widely used frameworks is QuEst [1], which extracts many features to describe the translation quality, such as fluency indicators, adequacy indicators, and translation complexity indicators, and it exploits a support vector regression algorithm to score the translation output. However, feature extraction requires part-of-speech analysis, syntactic parse, or semantic role labeling, yet these linguistic analyses relate to the target language types, which limits their application in other languages.

With the great success of deep learning that has been achieved in word distributed representations [2], language modeling [3], and machine translation [4]–[6], some researchers have proposed deep learning approaches to address the QE problem. Shah et al. [7]–[9] and Chen et al. [10] leverage neural features, such as word embedding and the recurrent neural network (RNN) language model, to improve the correlation between the automatic QE and human assessment [7]–[10]. Kim and colleagues established the-state-of-the-art bidirectional RNN encoder-decoder-based predictor and RNN-based estimator to estimate the translation quality [11]–[13]. Their approach achieved the best performances in the evaluation of the WMT17 sentence-level QE shared task [14], [15].

In this article, we propose to directly build and train a single, large end-to-end neural network for sentence-level QE, which reads a pair of source sentence and its machine translation and outputs a score indicating translation quality.

2. Related Work

QE methods exploiting deep learning can be classified into two categories. One is neural-aware QE, which integrates the neural features into a QE system. Shah et al. [7], [8] combine word embedding features and neural language model features generated from a continuous space language model training [16] with features extracted by QuEst [7], [8]. Shah et al. [9] further enrich the features with translation condition probabilities produced by a neural network machine translation system [9]. Chen et al. [10] exploit the continuous bag-of-words model [2] and RNN language model [3] to extract word embedding features and the cross-entropy feature for QE [10]. In these neural-aware QE approaches, the neural features effectively improve the system performance.

To more efficiently model, some approaches, called pure neural QE, build a neural network for QE. Kim and Lee [11] explored building RNNs for sentence-level QE, and the inputs of the networks are quality vectors produced by a bidirectional RNN encoder-decoder with attention mechanism [11]. Thus, the neural model consists of two sub-networks that are tuned separately. Kim et al. [12] further formalize the two sub-networks as an estimator and a predictor and extend the model to word-level QE and phrase-level QE [12]. To effectively train the RNNs, Kim et al. [13] exploited a stack propagation algorithm [17] to jointly tune the RNNs for word-level QE tasks, phrase-level QE tasks, and sentence-level QE tasks [13].

In line with this research, we combine the RNN and

Manuscript received January 24, 2018.
Manuscript revised April 8, 2018.
Manuscript publicized June 18, 2018.

[†]The authors are with the School of Computer Information Engineering, Jiangxi Normal University, Nanchang, 330022, China.

*This research has been funded by the Natural Science Foundation of China under Grant No.6146 2044, 6166 2031, and 6146 2045.

a) E-mail: mosesli@jxnu.edu.cn
b) E-mail: qingyuxiang@jxnu.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2018EDL8019

bidirectional RNN encoder-decoder with attention mechanism into a single, large neural network, called the unified neural network for sentence-level QE tasks (UNQE). Compared with the estimator and predictor approach [11]–[13], the proposed approach trains the networks together, rather than training the estimator and predictor separately. This means that the parameters of the bidirectional RNN encoder-decoder model are synchronously updated when fed QE instances, and the trained networks are more suited to sentence-level QE.

3. Methodology

The neural networks as depicted in Fig. 1 comprise two sub-networks: a bidirectional RNN encoder-decoder [6] and a QE RNN. The bidirectional RNN encoder-decoder is used to extract quality vectors given the translation context, which can be regarded as a feature extraction module, and the QE RNN uses the quality vector to predict the quality of the translation output, which can be regarded as a supervised regression module. The basic idea of combining these two sub-networks into one is to obtain a fine-grained network for QE tasks by jointly training to maximize the QE performance with QE samples.

3.1 Model Architecture

The bidirectional RNN encoder-decoder is a dominant sequence-to-sequence model widely used in machine translation [4]–[6]. The encoder maps an input source sentence $(x_1, \ldots, x_m)$ to a fixed-length vector. Given the vector, the decoder then generates a translation output $(y_1, \ldots, y_n)$ of symbols one word at a time. The generated conditional probability of each word can be written as follows:

$$p(y_j|\{y_1, \ldots, y_{j-1}\}, x) = g(y_{j-1}, s_{j-1}, c_j) = \frac{\exp(y_j^T W_o t_j)}{\sum_{k=1}^{K_y} \exp(y_k^T W_o t_j)} \quad (1)$$

where g denotes a nonlinear, potentially multi-layered function, c_j is the context vector, s_{j-1} is the hidden state of the RNN, $y_j \in R^{K_y \times 1}$ is the one-hot representation of the target word, K_y is the vocabulary size of the target language, $W_o \in R^{K_y \times d}$ is the weight matrix, $t_j \in R^{d \times 1}$ is the intermediate representation, and d is the dimension of the target language word.

The intermediate representation t_j can be inferred from the forward direction by the following formula:

$$t_j = \tanh(U_o s_{j-1} + V_o E y_{j-1} + C_o c_j) \quad (2)$$

where U_o, V_o, and C_o denote the model parameters. $E \in R^{d \times K_y}$ is the word embedding matrix for the target language.

To describe the translation quality regarding the target word-generated conditional probability, the quality vector is calculated as follows:

$$q_{y_j} = [(y_j^T W_o) \odot t_j^T]^T \quad (3)$$

where $\odot$ denotes an element-wise multiplication. The process of calculating the quality vector is depicted in Fig. 2.

Given the quality vectors, the most common approach is to use RNN as depicted in Fig. 3 such that

$$v_j = f(v_{j-1}, q_{y_j}) \quad (4)$$

where v_j is a hidden state at time j and f is a nonlinear function. Here, we use a GRU [5] as f to learn long-term

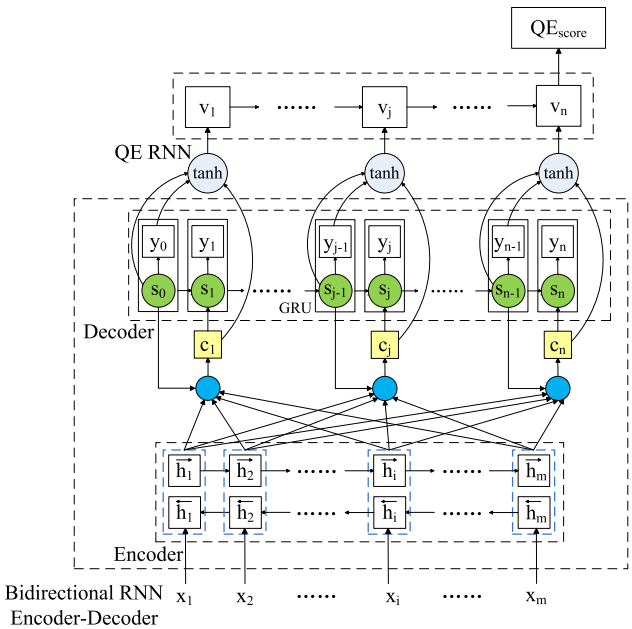

Fig. 1 An illustration of the proposed model architecture of the neural QE.

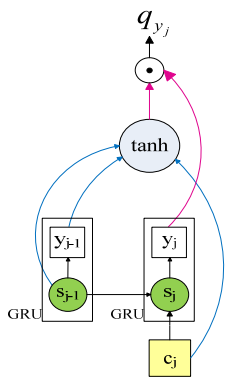

Fig. 2 An illustration of the calculation of quality vectors.

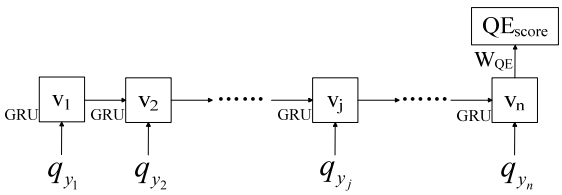

Fig. 3 An illustration of the RNN for QE.

dependencies of the translation outputs.

Because the last hidden state v_n sums up all the quality vectors of the translation output, it is used to predict the QE score as follows:

$$QE_{score} = W_{QE} \times v_n \tag{5}$$

where W_{QE} is a weight matrix. To make the QE score reasonable, we clip the score to range from 0 to 1 when this value is less than 0 or greater than 1. Note that our approach is different from that of Kim and Lee [11], in which the logistic sigmoid function was used to predict the QE score. The reason that we used the naïve score is that this approach had a better QE performance than using the logistic sigmoid function in the experiment.

3.2 Model Training

Assume that the training set for the QE task consists of N source sentences $x^{(n)}$, the translation outputs $y^{(n)}$, and the corresponding gold standard labels $HTER^{(n)}$ $(n = 1, \ldots, N)$. The training objective is to minimize the mean absolute error over the training data:

$$J(\theta) = \frac{1}{N} \sum_{n=1}^{N} |QE_{score}(x^{(n)}, y^{(n)}, \theta) - HTER^{(n)}| \tag{6}$$

where θ denotes the parameters of the network. In our approach, the quality vectors are intermediate variables that connected the bidirectional RNN encoder-decoder and the QE RNN. Thus, they changed with θ. This means that we can train the quality vectors for QE tasks. However, in the estimator and predictor approach that was proposed by Kim and Lee [11], the bidirectional RNN encoder-decoder trained for the machine translation task had constant quality vectors. This is why our approach is more reasonable.

Because the size of the training set for the QE task is too small to train the entire UNQE, the bidirectional RNN encoder-decoder and the QE RNN were pre-trained and initialized with the bilingual parallel corpus and QE training corpus, respectively. Then, the parameters of the networks were trained jointly with the QE training corpus.

4. Experimental Results

To test the performance of the proposed UNQE model, we conducted experiments on the WMT17 sentence-level QE task [15]. The statistics of the dataset are shown in Tables 1 and 2. The task involved estimating the quality of the translation outputs from English-to-German (en-de) and German-to-English (de-en) directions. The test set of the en-de direction consisted of a new test set, test set 2017, and an old test set used to measure the progress over the year, test set 2016.

4.1 Experimental Setting

To initialize the parameters of the bidirectional RNN encoder-decoder, the bilingual parallel corpus officially released by the WMT17 Translation task [15] was used, including Europarl v7, Common Crawl corpus, News Commentary v12, and Rapid corpus of EU press releases. Then, the networks were tuned using the training data released by the sentence-level QE task.

Table 1 Statistics of the en-de dataset of the WMT17 sentence-level QE task.

	Sentences	Words
Training data	23,000	404,198
Development data	1,000	19,487
Test data (test set 2016)	2,000	34,531
Test data (test set 2017)	2,000	35,577

Table 2 Statistics of the de-en dataset of the WMT17 sentence-level QE task.

	Sentences	Words
Training data	25,000	453,666
Development data	1,000	18,152
Test data	2,000	36,119

Pearson's correlation coefficient (Pearson's r) was used to evaluate the linear correlation between the automatic scores and the true HTER scores, while Spearman's rank correlation coefficient (Spearman's r) was used to evaluate whether the rankings generated from the automatic scores were similar to the true rankings generated from the true HTER scores. The higher the value of Pearson's r and Spearman's r, the more closely the model correlated with the human judgments.

The proposed UNQE model was contrasted with the traditional QE framework QuEst [1], the neural-aware QE models SHEF/QUEST-EMB [9] and JXNU/Emb+RNNLM +QuEst+SVM [10], and the pure neural QE model Predictor-Estimator [13]. The performances of the baseline models are cited from the officially released results of the WMT17 QE task [15].

4.2 Experimental Results

Tables 3–5 summarize the performances of the proposed UNQE models and the baseline models on the WMT17 sentence-level QE test set. On each table, the models were ordered by the value of Pearson's r.

We first compared the single proposed UNQE model with the single baseline models. The results showed that the UNQE model significantly outperformed the traditional QE approach QuEst and the two neural-aware QE models on each direction at the $p < 0.05$ level, as well as performed significantly better than the Predictor-Estimator model with the stack propagation algorithm at the en-de translation direction (including test set 2016 and test set 2017) at the $p < 0.05$ level, and consistently outperformed the Predictor-Estimator model at the de-en translation directions. The UNQE model gained approximately 0.014–0.409 improvement in the Pearson's r value and 0.001–0.307 improvement in the Spearman's r value over the baseline models. The results confirmed that the UNQE model that trained the net-

Table 3 Performance of the models on the WMT17 sentence-level QE en-de test set (test set 2016).

Model	Scoring	Ranking
	Pearson's r	Spearman's r
UNQE-Ensemble	**0.717**	**0.746**
Predictor-Estimator-Ensemble [13]	0.714	0.736
UNQE	0.708	0.737
UNQE with sigmoid	0.702	0.723
PredictorEstimator [13]	0.686	0.707
JXNU/Emb+RNNLM+QuEst+SVM [10]	0.527	0.552
SHEF/QUEST-EMB [9]	0.499	0.527
QuEst [1]	0.399	0.438

Table 4 Performance of the models on the WMT17 sentence-level QE en-de test set (test set 2017).

Model	Scoring	Ranking
	Pearson's r	Spearman's r
UNQE-Ensemble	**0.710**	**0.740**
UNQE	0.700	0.732
PredictorEstimator-Ensemble [13]	0.695	0.725
UNQE with sigmoid	0.685	0.717
PredictorEstimator [13]	0.673	0.703
JXNU/Emb+RNNLM+QuEst+SVM [10]	0.522	0.545
SHEF/QUEST-EMB [9]	0.496	0.513
QuEst [1]	0.397	0.425

Table 5 Performance of the models on the WMT17 sentence-level QE de-en test set.

Model	Scoring	Ranking
	Pearson's r	Spearman's r
UNQE-Ensemble	**0.738**	0.681
UNQE	0.729	0.671
PredictorEstimator-Ensemble [13]	0.728	**0.690**
PredictorEstimator [13]	0.715	0.670
UNQE with sigmoid	0.714	0.626
SHEF/QUEST-EMB [9]	0.558	0.560
JXNU/Emb+RNNLM+QuEst+SVM [10]	0.531	0.520
QuEst [1]	0.441	0.450

work is better than the predictor-estimator model that separately trained two sub-networks. The reason is that the bidirectional RNN encoder-decoder were trained regarding the QE training samples in the UNQE model. Therefore, the sub-networks generated more accurate features for the QE task.

Next, we ensemble six likelihood-trained UNQE models by changing the dimension of the word embedding from 500 to 700 and the dimension of the RNN from 100 to 200 at intervals of 50, and we compared the results to the predictor-estimator model, which also reported ensemble results. The results showed that the ensemble model, UNQE-Ensemble, further improved the correlation between the automatic scores and the true labels compared to the single model, UNQE, on each direction. In addition, in most cases, the performance of the UNQE-Ensemble model was superior to the-state-of-art ensemble model, the Predictor-Estimator-Ensemble that included 15 single models [13].

4.3 Effect of the Naïve Score Approach

Tables 3–5 also provide the QE system performance of the model using the naïve score defined in formula (5) (UNQE) and the model using the logistic sigmoid function (UNQE with sigmoid). The results showed that the model using the naïve score consistently outperformed the model using the logistic sigmoid function.

The reason may be that most of the QE score values which employed the naïve score approach are already among [0, 1], however, if the nonlinear logistic sigmoid activation function was employed, it will reduce the QE score value of the translation outputs. For example, given a source sentence "*Eine fungizide Wirkung von Caspofungin wurde gegen Candida-Hefen nachgewiesen.*" and the corresponding human reference "*One fungicidal action of caspofungin has been demonstrated against candida yeast.*" when estimating the quality of the translation output "*One fungicidal action of caspofungin has been demonstrated to Candida-Hefen.*" in the de-en direction, it shows that the values (0.247) generated by the naïve scores approach are closer to the true HTER score (0.250) than the values (0.190) generated by the sigmoid function.

5. Conclusion

We introduced a unified, end-to-end neural network for sentence-level QE, which was built and trained as a whole. On the WMT17 sentence-level QE task, the proposed single model consistently outperformed the best single participated model. In future work, we would like to simplify the structure of the networks to obtain better results.

References

[1] L. Specia, K. Shah, J.G.C. de Souza, and T. Cohn, "QuEst – A translation quality estimation framework," Proc. ACL, Sofia, Bulgaria, pp.79–84, 2013.

[2] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," Advances in Neural Information Processing Systems 26, pp.3111–3119, 2013.

[3] T. Mikolov, M. Karafiat, L. Burget, J.H. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," Proc. INTERSPEECH, Makuhari, Chiba, Japan, pp.1045–1048, 2010.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," Proc. ICLR, Hilton San Diego Resort & Spa, pp.1–15, 2015.

[5] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," Proc. EMNLP, Doha, Qatar, pp.1724–1734, 2014.

[6] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A.V.M. Barone, J. Mokry, and M. Nădejde, "Nematus: A toolkit for neural machine translation," Proc. EACL, Valencia, Spain, pp.65–68, 2017.

[7] K. Shah, V. Logacheva, G.H. Paetzold, F. Blain, D. Beck, F. Bougares, and L. Specia, "SHEF-NN: Translation quality estimation with neural networks," Proc. WMT, Lisbon, Portugal, pp.342–347,

2015.

[8] K. Shah, R.W.M. Ng, F. Bougares, and L. Specia, "Investigating continuous space language models for machine translation quality estimation," Proc. EMNLP, Lisbon, Portugal, pp.1073–1078, 2015.

[9] K. Shah, F. Bougares, L. Barrault, and L. Specia, "SHEF-LIUM-NN: Sentence level quality estimation with neural network features," Proc. WMT, Berlin, Germany, pp.838–842, 2016.

[10] Z. Chen, Y. Tan, C. Zhang, Q. Xiang, L. Zhang, M. Li, and M. Wang, "Improving machine translation quality estimation with neural network features," Proc. WMT, Denmark, pp.551–555, 2016.

[11] H. Kim and J.-H. Lee, "A recurrent neural networks approach for estimating the quality of machine translation output," Proc. NAACL-HLT, San Diego, California, pp.494–498, 2016.

[12] H. Kim, H.-Y. Jung, H. Kwon, J.-H. Lee, and S.-H. Na, "Predictor-estimator: Neural quality estimation based on target word prediction for machine translation," ACM TALIIP, vol.17, no.1, pp.1–22, 2017.

[13] H. Kim, J.-H. Lee, and S.-H. Na, "Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation," Proc. WMT, Copenhagen, Denmark, pp.562–568, 2017.

[14] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A.J. Yepes, P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Névéol, M. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor, and M. Zampieri, "Findings of the 2016 conference on machine translation," Proc. WMT, Berlin, Germany, pp.131–198, 2016.

[15] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, C. Monz, M. Negri, M. Post, R. Rubino, L. Specia, and M. Turchi, "Findings of the 2017 conference on machine translation," Proc. WMT, Copenhagen, Denmark, pp.169–214, 2017.

[16] H. Schwenk, "Continuous space translation models for phrase-based statistical machine translation," Proc. COLING, Mumbai, India, pp.1071–1080, 2012.

[17] Y. Zhang and D. Weiss, "Stack-propagation: Improved representation learning for syntax," Proc. ACL, Berlin, Germany, pp.1557–1566, 2016.