Contextual Encoding for Translation Quality Estimation

Junjie Hu, Wei-Cheng Chang, Yuexin Wu, Graham Neubig
Language Technologies Institute, Carnegie Mellon University
{junjieh, wchang2, yuexinw, gneubig}@cs.cmu.edu

Abstract

The task of word-level quality estimation (QE) consists of taking a source sentence and machine-generated translation, and predicting which words in the output are correct and which are wrong. In this paper, propose a method to effectively encode the local and global contextual information for each target word using a three-part neural network approach. The first part uses an embedding layer to represent words and their part-of-speech tags in both languages. The second part leverages a one-dimensional convolution layer to integrate local context information for each target word. The third part applies a stack of feed-forward and recurrent neural networks to further encode the global context in the sentence before making the predictions. This model was submitted as the CMU entry to the WMT2018 shared task on QE, and achieves strong results, ranking first in three of the six tracks.[1]

1 Introduction

Quality estimation (QE) refers to the task of measuring the quality of machine translation (MT) system outputs without reference to the gold translations (Blatz et al., 2004; Specia et al., 2013). QE research has grown increasingly popular due to the improved quality of MT systems, and potential for reductions in post-editing time and the corresponding savings in labor costs (Specia, 2011; Turchi et al., 2014). QE can be performed on multiple granularities, including at word level, sentence level, or document level. In this paper, we focus on quality estimation at word level, which is framed as the task of performing binary classification of translated tokens, assigning "OK" or "BAD" labels.

[1]Our software is available at https://github.com/junjiehu/CEQE.

Early work on this problem mainly focused on hand-crafted features with simple regression/classification models (Ueffing and Ney, 2007; Biçici, 2013). Recent papers have demonstrated that utilizing recurrent neural networks (RNN) can result in large gains in QE performance (Martins et al., 2017). However, these approaches encode the context of the target word by merely concatenating its left and right context words, giving them limited ability to control the interaction between the local context and the target word.

In this paper, we propose a neural architecture, Context Encoding Quality Estimation (CEQE), for better encoding of context in word-level QE. Specifically, we leverage the power of both (1) convolution modules that automatically learn local patterns of surrounding words, and (2) hand-crafted features that allow the model to make more robust predictions in the face of a paucity of labeled data. Moreover, we further utilize stacked recurrent neural networks to capture the long-term dependencies and global context information from the whole sentence.

We tested our model on the official benchmark of the WMT18 word-level QE task. On this task, it achieved highly competitive results, with the best performance over other competitors on English-Czech, English-Latvian (NMT) and English-Latvian (SMT) word-level QE task, and ranking second place on English-German (NMT) and German-English word-level QE task.

2 Model

The QE module receives as input a tuple $\langle s, t, \mathcal{A} \rangle$, where $s = s_1, \ldots, s_M$ is the source sentence, $t = t_1, \ldots, t_N$ is the translated sentence, and $\mathcal{A} \subseteq \{(m, n) | 1 \leq m \leq M, 1 \leq n \leq N\}$ is a set of word alignments. It predicts as output a sequence $\hat{y} = y_1, \ldots, y_N$, with each $y_i \in \{\text{BAD}, \text{OK}\}$. The

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 801–806

https://doi.org/10.18653/v1/W18-6463

overall architecture is shown in Figure 1

CEQE consists of three major components: (1) embedding layers for words and part-of-speech (POS) tags in both languages, (2) convolution encoding of the local context for each target word, and (3) encoding the global context by the recurrent neural network.

2.1 Embedding Layer

Inspired by (Martins et al., 2017), the first embedding layer is a vector representing each target word t_j obtained by concatenating the embedding of that word with those of the aligned words $s_{\mathcal{A}(:,t_j)}$ in the source. If a target word is aligned to multiple source words, we average the embedding of all the source words, and concatenate the target word embedding with its average source embedding. The immediate left and right contexts for source and target words are also concatenated, enriching the local context information of the embedding of target word t_j. Thus, the embedding of target word t_j, denoted as $\mathbf{x}_j$, is a $6d$ dimensional vector, where d is the dimension of the word embeddings. The source and target words use the same embedding parameters, and thus identical words in both languages, such as digits and proper nouns, have the same embedding vectors. This allows the model to easily identify identical words in both languages. Similarly, the POS tags in both languages share the same embedding parameters. Table 1 shows the statistics of the set of POS tags over all language pairs.

Language Pairs	Source	Target
En-De (SMT)	50	57
En-De (NMT)	49	58
De-En	58	50
En-Lv (SMT)	140	38
En-Lv (NMT)	167	43
En-Cz	440	57

Table 1: Statistics of POS tags over all language pairs

2.2 One-dimensional Convolution Layer

The main difference between the our work and the neural model of Martins et al. (2017) is the one-dimensional convolution layer. Convolutions provide a powerful way to extract local context features, analogous to implicitly learning n-gram features. We now describe this integral part of our model.

After embedding each word in the target sentence $\{t_1, \ldots, t_j, \ldots, t_N\}$, we obtain a matrix of embeddings for the target sequence,

$$\mathbf{x}_{1:N} = \mathbf{x}_1 \oplus \mathbf{x}_2 \ldots \oplus \mathbf{x}_N,$$

where $\oplus$ is the column-wise concatenation operator. We then apply one-dimensional convolution (Kim, 2014; Liu et al., 2017) on $\mathbf{x}_{1:N}$ along the target sequence to extract the local context of each target word. Specifically, a one-dimensional convolution involves a filter $\mathbf{w} \in \mathbb{R}^{hk}$, which is applied to a window of h words in target sequence to produce new features.

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b),$$

where $b \in \mathbb{R}$ is a bias term and f is some functions. This filter is applied to each possible window of words in the embedding of target sentence $\{\mathbf{x}_{1:h}, \mathbf{x}_{2:h+1}, \ldots, \mathbf{x}_{N-h+1:N}\}$ to produce features

$$\mathbf{c} = [c_1, c_2, \ldots, c_{N-h+1}].$$

By the padding proportionally to the filter size h at the beginning and the end of target sentence, we can obtain new features $\mathbf{c}_{pad} \in \mathbb{R}^N$ of target sequence with output size equals to input sentence length N. To capture various granularities of local context, we consider filters with multiple window sizes $\mathcal{H} = \{1, 3, 5, 7\}$, and multiple filters $n_f = 64$ are learned for each window size.

The output of the one-dimensional convolution layer, $C \in \mathbb{R}^{N \times |\mathcal{H}| \cdot n_f}$, is then concatenated with the embedding of POS tags of the target words, as well as its aligned source words, to provide a more direct signal to the following recurrent layers.

2.3 RNN-based Encoding

After we obtain the representation of the source-target word pair by the convolution layer, we follow a similar architecture as (Martins et al., 2017) to refine the representation of the word pairs using feed-forward and recurrent networks.

1. Two feed-forward layers of size 400 with rectified linear units (ReLU; Nair and Hinton (2010));

2. One bi-directional gated recurrent unit (Bi-GRU; Cho et al. (2014)) layer with hidden size 200, where the forward and backward hidden states are concatenated and further normalized by layer normalization (Ba et al., 2016).

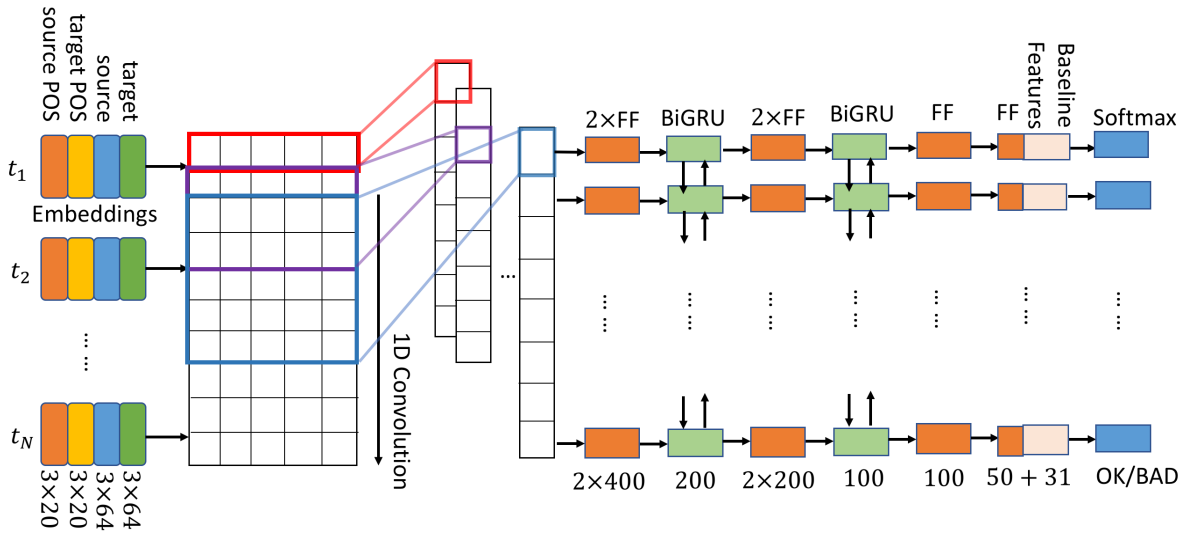

Figure 1: The architecture of our model, with the convolutional encoder on the left, and stacked RNN on the right.

Category	Description
Binary	target word is a stopword
Binary	target word is a punctuation mark
Binary	target word is a proper noun
Binary	target word is a digit
Float	backoff behavior of ngram w_{i-2} w_{i-1} w_i (w_i is the target word)
Float	backoff behavior of ngram w_{i-1} w_i w_{i+1}
Float	backoff behavior of ngram w_i w_{i+1} w_{i+2}
One-hot	highest order of ngram that includes target word and its left context
One-hot	highest order of ngram that includes target word and its right context
One-hot	highest order of ngram that includes source word and its left context
One-hot	highest order of ngram that includes source word and its right context

Table 2: Baseline Features

3. Two feed-forward layers of hidden size 200 with rectified linear units;

4. One BiGRU layer with hidden size 100 using the same configuration of the previous BiGRU layer;

5. Two feed-forward layers of size 100 and 50 respectively with ReLU activation.

We concatenate the 31 baseline features extracted by the Marmot[2] toolkit with the last 50 feed-forward hidden features. The baseline features are listed in Table 2. We then apply a softmax layer on the combined features to predict the binary labels.

3 Training

We minimize the binary cross-entropy loss between the predicted outputs and the targets. We train our neural model with mini-batch size 8 using Adam (Kingma and Ba, 2015) with learning rate 0.001 and decay the learning rate by multiplying 0.75 if the F1-Multi score on the validation set decreases during the validation. Gradient norms are clipped within 5 to prevent gradient explosion for feed-forward networks or recurrent neural networks. Since the training corpus is rather small, we use dropout (Srivastava et al., 2014) with probability 0.3 to prevent overfitting.

4 Experiment

We evaluate our CEQE model on the WMT2018 Quality Estimation Shared Task[3] for word-level English-German, German-English, English-Czech, and English-Latvian QE. Words in all languages are lowercased. The evaluation metric is the multiplication of F1-scores for the "OK" and "BAD" classes against the true labels. F1-score is the harmonic mean of precision and recall. In Table 3, our model achieves the best performance on three out of six test sets in the WMT 2018 word-level QE shared task.

4.1 Ablation Analysis

In Table 4, we show the ablation study of the features used in our model on English-German, German-English, and English-Czech. For each

[2]https://github.com/qe-team/marmot

[3]http://statmt.org/wmt18/quality-estimation-task.html

Language Pairs	F1-BAD	F1-OK	F1-Multi	Rank
En-De (SMT)	0.5075	0.8394	0.4260	3
En-De (NMT)	0.3565	0.8827	0.3147	2
De-En	0.4906	0.8640	0.4239	2
En-Lv (SMT)	0.4211	0.8592	0.3618	1
En-Lv (NMT)	0.5192	0.8268	0.4293	1
En-Cz	0.5882	0.8061	0.4741	1

Table 3: Best performance of our model on six datasets in the WMT2018 word-level QE shared task on the leader board (updated on July 27th 2018)

language pair, we show the performance of CEQE without adding the corresponding components specified in the second column respectively. The last row shows the performance of the complete CEQE with all the components. As the baseline features released in the WMT2018 QE Shared Task for English-Latvian are incomplete, we train our CEQE model without using such features. We can glean several observations from this data:

1. Because the number of "OK" tags is much larger than the number of "BAD" tags, the model is easily biased towards predicting the "OK" tag for each target word. The F1-OK scores are higher than the F1-BAD scores across all the language pairs.

2. For German-English, English Czech, and English-German (SMT), adding the baseline features can significantly improve the F1-BAD scores.

3. For English-Czech, English-German (SMT), and English-German (NMT), removing POS tags makes the model more biased towards predicting "OK" tags, which leads to higher F1-OK scores and lower F1-BAD scores.

4. Adding the convolution layer helps to boost the performance of F1-Multi, especially on English-Czech and English-Germen (SMT) tasks. Comparing the F1-OK scores of the model with and without the convolution layer, we find that adding the convolution layer help to boost the F1-OK scores when translating from English to other languages, i.e., English-Czech, English-German (SMT and NMT). We conjecture that the convolution layer can capture the local information more effectively from the aligned source words in English.

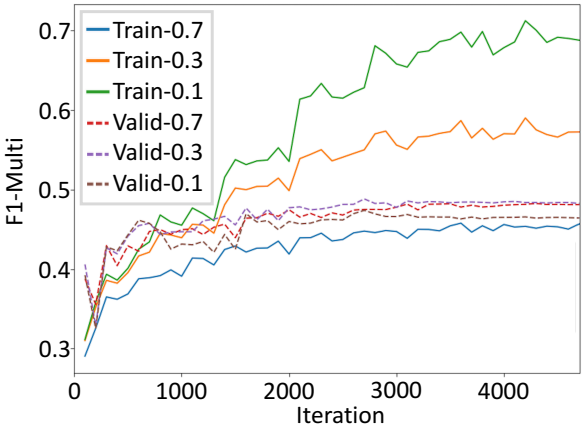

Figure 2: Effect of the dropout rate during training.

5 Case Study

Table 5 shows two examples of quality prediction on the validation data of WMT2018 QE task for English-Czech. In the first example, the model without POS tags and baseline features is biased towards predicting "OK" tags, while the model with full features can detect the reordering error. In the second example, the target word "panelu" is a variant of the reference word "panel". The target word "znaky" is the plural noun of the reference "znak". Thus, their POS tags have some subtle differences. Note the target word "zmnit" and its aligned source word "change" are both verbs. We can observe that POS tags can help the model capture such syntactic variants.

5.1 Sensitivity Analysis

During training, we find that the model can easily overfit the training data, which yields poor performance on the test and validation sets. To make the model more stable on the unseen data, we apply dropout to the word embeddings, POS embeddings, vectors after the convolutional layers and the stacked recurrent layers. In Figure 2, we examine the accuracies dropout rates in $[0.1, 0.3, 0.7]$. We find that adding dropout alleviates overfitting issues on the training set. If we reduce the dropout rate to 0.1, which means randomly setting some values to zero with probability 0.1, the training F1-Multi increases rapidly and the validation F1-multi score is the lowest among all the settings. Preliminary results proved best for a dropout rate of 0.3, so we use this in all the experiments.

Language Pairs	Method	F1-BAD	F1-OK	F1-Multi
De-En	- (Convolution + POS + features)	0.4774	0.8680	0.4144
	- (POS + features)	0.4948	0.8474	0.4193
	- features	0.5095	**0.8735**	0.4450
	- POS	0.4906	0.8640	0.4239
	CEQE	**0.5233**	0.8721	**0.4564**
En-Cz	- (Convolution + POS + features)	0.5748	0.7622	0.4381
	- (POS + features)	0.5628	0.8000	0.4502
	- features	0.5777	0.7997	0.4620
	- POS	0.5192	**0.8268**	0.4293
	CEQE	**0.5884**	0.7991	**0.4702**
En-De (SMT)	- (Convolution + POS + features)	0.4677	0.8038	0.3759
	- (POS + features)	0.4768	0.8166	0.3894
	- features	0.4902	0.8230	0.4034
	- POS	0.5047	**0.8431**	0.4255
	CEQE	**0.5075**	0.8394	**0.4260**
En-De (NMT)	- (Convolution + POS + features)	0.3545	0.8396	0.2976
	- (POS + features)	0.3404	0.8752	0.2979
	- features	**0.3565**	0.8827	**0.3147**
	- POS	0.3476	**0.8948**	0.3111
	CEQE	0.3481	0.8835	0.3075

Table 4: Ablation study on the WMT18 Test Set

6 Conclusion

In this paper, we propose a deep neural architecture for word-level QE. Our framework leverages a one-dimensional convolution on the concatenated word embeddings of target and its aligned source words to extract salient local feature maps. In additions, bidirectional RNNs are applied to capture temporal dependencies for better sequence prediction. We conduct thorough experiments on four language pairs in the WMT2018 shared task. The proposed framework achieves highly competitive results, outperforms all other participants on English-Czech and English-Latvian word-level, and is second place on English-German, and German-English language pairs.

Acknowledgements

The authors thank Andre Martins for his advice regarding the word-level QE task.

This work is sponsored by Defense Advanced Research Projects Agency Information Innovation Office (I2O). Program: Low Resource Languages for Emergent Incidents (LORELEI). Issued by DARPA/I2O under Contract No. HR0011-15-C0114. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Ergun Biçici. 2013. Referential translation machines for quality estimation. In *Proceedings of the eighth workshop on statistical machine translation*, pages 343–351.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 315. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Source	specify the scope of blending options :
MT	určete rozsah prolnutí voleb :
Reference	určete rozsah voleb prolnutí :
no POS & features	určete rozsah prolnutí voleb :
CEQE	určete rozsah prolnutí voleb :
Source	use the Character panel and Paragraphs panel to change the appearance of text .
MT	pomocí panelu znaky a odstavce , chcete - li změnit vzhled textu .
Reference	použijte panel znak a panel odstavce , chcete - li změnit vzhled textu .
no POS & features	pomocí panelu znaky a odstavce , chcete - li zmnit vzhled textu .
CEQE	pomocí panelu znaky a odstavce , chcete - li změnit vzhled textu .

Table 5: Examples on WMT2018 validation data. The source and translated sentences, the reference sentences, the predictions of the CEQE without and with POS tags and baseline features are shown. Words predicted as OK are shown in green, those predicted as BAD are shown in red, the difference between the translated and reference sentences are shown in blue.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124. ACM.

André Martins, Marcin Junczys-Dowmunt, Fabio Kepler, Ramón Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics*, 5:205–218.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Lucia Specia. 2011. Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pages 73–80.

Lucia Specia, Kashif Shah, Jose GC Souza, and Trevor Cohn. 2013. Quest-a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Marco Turchi, Antonios Anastasopoulos, José GC de Souza, and Matteo Negri. 2014. Adaptive quality estimation for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 710–720.

Nicola Ueffing and Hermann Ney. 2007. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40.