A Recurrent Neural Networks Approach for Estimating the Quality of Machine Translation Output

Hyun Kim
Creative IT Engineering,
Pohang University of Science and Technology (POSTECH),
Pohang, Republic of Korea
hkim.postech@gmail.com

Jong-Hyeok Lee
Computer Science and Engineering,
Pohang University of Science and Technology (POSTECH),
Pohang, Republic of Korea
jhlee@postech.ac.kr

Abstract

This paper presents a novel approach using recurrent neural networks for estimating the quality of machine translation output. A sequence of vectors made by the prediction method is used as the input of the final recurrent neural network. The prediction method uses bi-directional recurrent neural network architecture both on source and target sentence to fully utilize the bi-directional quality information from source and target sentence. Our experiments show that the proposed recurrent neural networks approach achieves a performance comparable to the existing state-of-the-art models for estimating the sentence-level quality of English-to-Spanish translation.

1 Introduction

Estimating the quality of machine translation output, called *quality estimation* (QE) (Specia et al., 2009; Blatz et al., 2004), is to predict quality scores/categories for unseen machine-translated sentences without reference translations at various granularity levels (sentence-level/word-level/document-level). Quality estimation is of growing importance in the field of machine translation (MT) since MT systems are widely used and the quality of each machine-translated sentence is able to vary considerably.

Previous research on QE, addressed as a regression/classification problem to compute quality scores/categories, has mainly focused on feature extraction and feature selection. Feature extraction is to find the relevant features, such as baseline features (Specia et al., 2013) and latent semantic indexing (LSI) based features (Langlois, 2015), capturing various aspects of quality from source and target sentences[1] and external resources. Feature selection is to select the best features by using selection algorithms, such as Gaussian processes (Shah et al., 2015) and heuristic (González-Rubio et al., 2013), among already extracted features. Finding desirable features has played a key role in the QE research.

In this paper we present a recurrent neural networks approach for estimating the quality of machine translation output at sentence level, which does not require manual effort for finding the best relevant features. The remainder of this paper is organized as follows. In Section 2, we propose a recurrent neural networks approach using a sequence of vectors made by the prediction method as input for quality estimation. And we describe the prediction method using bi-directional recurrent neural networks architecture in Section 3. In Section 4, we report evaluation results, and conclude our paper in Section 5.

2 Recurrent Neural Networks Approach for Estimating Quality Score

Because recurrent neural networks (RNNs) have the strength for handling sequential data (Goodfellow et al., 2015), we apply RNNs to estimate the quality score of translation.

The input of the final RNN is a sequence of vectors that have quality information about whether tar-

[1] In this paper, a 'target sentence' means the machine-translated sentence from a source sentence.

Proceedings of NAACL-HLT 2016, pages 494–498,
San Diego, California, June 12-17, 2016.

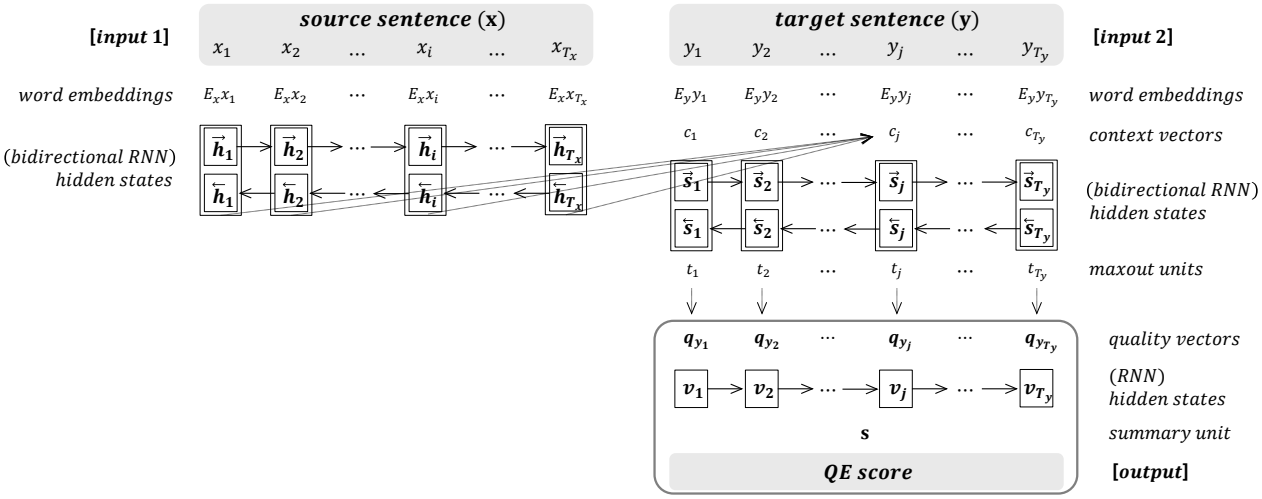

Figure 1: An illustration of the proposed recurrent neural networks model for quality estimation

get words in a target sentence are properly translated from a source sentence. We will refer to this sequence of vectors as *quality vectors* $(q_{y_1}, \ldots, q_{y_{T_y}})$. Each quality vector q_{y_j}[2] has the quality information about how well a target word y_j in a target sentence $\mathbf{y} = (y_1, \ldots, y_{T_y})$ is translated from a source sentence[3] $\mathbf{x} = (x_1, \ldots, x_{T_x})$. Quality vectors are generated from the prediction method (of Section 3).

To predict a quality estimation score (QE score) as an HTER score (Snover et al., 2006) in [0,1] for each target sentence, a logistic sigmoid function is used such that

$$\begin{aligned} &\mathrm{QE\,score}(\mathbf{y}, \mathbf{x}) \\ &= \mathrm{QE\,score}'(q_{y_1}, \ldots, q_{y_{T_y}}) \\ &= \sigma(W_{QE}^{\top}\, \mathbf{s}) \end{aligned} \tag{1}$$

where $\mathbf{s}$ is a summary unit of the whole quality vectors and $W_{QE} \in \mathbb{R}^r$. r is the dimensionality of summary unit.

To get the summary unit $\mathbf{s}$, the hidden state v_j employing p gated hidden units for the target word y_j is computed by

$$v_j = f(q_{y_j}, v_{j-1})\,. \tag{2}$$

The gated hidden unit (Cho et al., 2014) for the activation function f is used to learn long-term dependencies of translation qualities for target words. We consider the QE score as the integrated/condensed value reflecting the sequential quality information of sequential target words. Because the last hidden state v_{T_y} is a summary of the sequential quality vectors, we fix the summary unit $\mathbf{s}$ to the last hidden state v_{T_y}.

3 Prediction method using Bi-directional RNN Architecture to Make Quality Vectors

In this section, we detail the ways to get the quality vectors $(q_{y_1}, \ldots, q_{y_{T_y}})$ for computing QE score.

Since the training data for QE[4] are not enough to use a neural networks approach for making quality vectors, we use an alternative based on large-scale parallel corpora such as Europarl. We modify the word prediction method of RNN Encoder-Decoder (Cho et al., 2014) using parallel corpora to make the quality vectors.

In subsection 3.1, we describe the underlying word prediction method of RNN Encoder-Decoder. We i) extend the prediction method to use the additional backward RNN architecture on target sentence in subsection 3.2 and ii) modify to get the quality vectors $(q_{y_1}, \ldots, q_{y_{T_y}})$ in subsection 3.3.

[2] $1 \leqq j \leqq T_y$ where T_y is the length of target sentence.

[3] Source(Target) sentence consists of 1-of-$K_x(K_y)$ coded word vectors. $K_x(K_y)$ is the vocabulary sizes of source(target) language.

[4] These data, provided in WMT Quality Estimation Shared Task, consist of source sentences, target sentences, and quality scores.

Figure 1 is the graphical illustration of the proposed RNNs approach.

3.1 Word Prediction Method of RNN Encoder-Decoder

RNN Encoder-Decoder proposed by Cho et al. (2014) is able to predict the target word y_j given a source sentence $\mathbf{x}$ and all preceding target words $\{y_1, ..., y_{j-1}\}$ by using a softmax function. And it is extended by Bahdanau et al. (2015) to use information of relevant source words for predicting the target word y_j such that

$$p(y_j|\{y_1, ..., y_{j-1}\}, \mathbf{x}) = g(y_{j-1}, \vec{s}_{j-1}, c_j)\,. \quad (3)$$

g is a nonlinear function predicting the probability of y_j. $\vec{s}_{j-1}$ is the hidden state of the forward RNN on target sentence and contains information of preceding target words $\{y_1, \dots, y_{j-1}\}$. c_j is the context vector which means relevant parts of source sentence associated with the target word y_j. $\vec{s}_{j-1}$ and y_{j-1} are related to all preceding target words $\{y_1, ..., y_{j-1}\}$, and c_j is related to $\mathbf{x}$ in the word prediction function of (3).

3.2 Additional Backward RNN Architecture on Target Sentence

Bahdanau et al. (2015) introduce bi-directional RNN architecture only on source sentence to extend RNN Encoder-Decoder. In our proposed QE model, bi-directional RNN architecture is used both on source and target sentence. By applying bi-directional RNN architecture both on source and target sentence, we can fully and bi-directionally utilize source and target sentence for predicting target words, such that

$$\begin{aligned} &p(y_j|\mathbf{y}_{\not\ni y_j}, \mathbf{x}) \\ &\quad = g([y_{j-1}; y_{j+1}], [\vec{s}_{j-1}; \overleftarrow{s}_{j+1}], c_j) \\ &\quad = \frac{\exp(y_j^\top W_{o_1} W_{o_2} t_j)}{\sum_{k=1}^{K_y} \exp(y_k^\top W_{o_1} W_{o_2} t_j)}\,, \end{aligned} \quad (4)$$

which is the extended version of (3) using the additional backward RNN architecture.[5]

To reflect further all following target words $\{y_{j+1}, \dots, y_{T_y}\}$ when predicting the target word y_j, the hidden state $\overleftarrow{s}_{j+1}$ of the backward RNN and the next target word y_{j+1} are added. $[\vec{s}_{j-1}; \overleftarrow{s}_{j+1}]$ and $[y_{j-1}; y_{j+1}]$ are related to $\mathbf{y}_{\not\ni y_j}$[6], and c_j is related to $\mathbf{x}$ in the word prediction function of (4).

$W_{o_1} \in \mathbb{R}^{K_y \times q}$ and $W_{o_2} \in \mathbb{R}^{q \times l}$ are weight matrices of softmax function. K_y is the vocabulary sizes of target language and q is the dimensionality of quality vectors. l is the dimensionality of maxout units such that

$$t_j = [max\{\tilde{t}_{j,2k-1}, \tilde{t}_{j,2k}\}]_{k=1,\dots,l}^\top\,, \quad (5)$$

where $\tilde{t}_{j,k}$ is the k-th element of a vector $\tilde{t}_j$. And

$$\tilde{t}_j = S'_o[\vec{s}_{j-1}; \overleftarrow{s}_{j+1}] + V'_o[E_{\mathbf{y}} y_{j-1}; E_y y_{j+1}] + C_o c_j, \quad (6)$$

where $S'_o \in \mathbb{R}^{2l \times 2n}$, $V'_o \in \mathbb{R}^{2l \times 2m}$, and $C_o \in \mathbb{R}^{2l \times 2n}$. $E_{\mathbf{y}} \in \mathbb{R}^{m \times K_y}$ is the word embedding matrix on target sentence. m and n are the dimensionality of word embedding and hidden states of forward and backward RNNs. The hidden state $\overleftarrow{s}_{j+1}$ of the backward RNN and next target word y_{j+1} are used in (6).[7]

From the extended prediction method of (4), the probability of the target word y_j is computed by using information of relevant source words in source sentence $\mathbf{x}$ and all target words $\mathbf{y}_{\not\ni y_j}$ surrounding the target word y_j in target sentence.

3.3 Quality Vectors on Target Sentence

Word prediction method predicts the probability of target words as a number between 0 and 1. But we want to get quality vectors of q-dimensionality which have the more intrinsic quality information for target words.

To make quality vectors, we regard that the probability of the target word y_j involves the quality information about whether the target word y_j in target sentence is properly translated from source sentence. Thus, by decomposing the softmax function[8] of (4),

[5] The additional backward RNN on target sentence use the context vectors shared by the forward RNN on target sentence.

[6] $\mathbf{y}_{\not\ni y_j} = \{y_1, \dots, y_{j-1}, y_{j+1}, \dots, y_{T_y}\}$

[7] Original $\tilde{t}_j$ (Bahdanau et al., 2015) is
$$\tilde{t}_j = S_o \vec{s}_{j-1} + V_o E_{\mathbf{y}} y_{j-1} + C_o c_j.$$

[8] In this softmax function, the bias term is not used for the simplicity of deriving the quality vectors. Generally, bias terms are visually omitted in other equations to make the equations uncluttered.

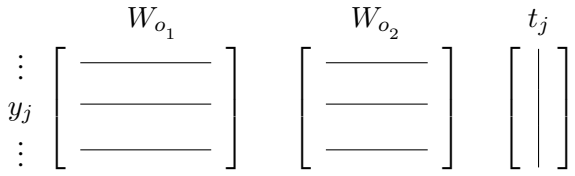

Figure 2: Weight matrices (W_{o_1} and W_{o_2}) of softmax function and maxout unit t_j for the target word y_j

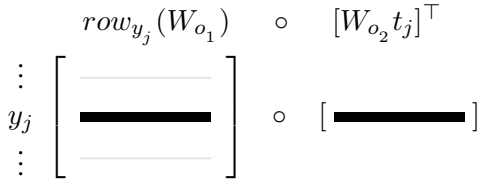

Figure 3: The ways of computing the quality vector q_{y_j} ($\circ$ is an element-wise multiplication)

the quality vector q_{y_j} for the target word y_j is computed by

$$q_{y_j} = \left[\; row_{y_j}(W_{o_1}) \circ [W_{o_2}t_j]^\top \; \right]^\top, \quad (7)$$

where $\circ$ is an element-wise multiplication. All of quality information about possible K_y target words at position j of target sentence is encoded in t_j. Thus, by decoding t_j, we are able to get quality vector q_{y_j} for the target word $y_j \in \mathbb{R}^{K_y}$ at position j of target sentence. Figure 2 and 3 show the ways to compute the quality vector q_{y_j}.

4 Experiments

The proposed RNNs approach was evaluated on the WMT15 Quality Estimation Shared Task[9] at sentence level of English-Spanish.

We trained[10] the proposed model through a two-step process. First, by using English-Spanish parallel corpus of Europarl v7 (Koehn, 2005), we trained bi-directional RNNs having 1000 hidden units on source and target sentence to make quality vectors. Next, by using the training set of WMT15 QE task, to predicte QE scores we trained the final RNN that use the quality vectors generated in previous step as the input and have 100 hidden units.

System ID	**MAE** ↓	RMSE ↓
• RTM-DCU/RTM-FS+PLS-SVR	0.1325	0.1748
• LORIA/17+LSI+MT+FILTRE	0.1334	0.1735
• RTM-DCU/RTM-FS-SVR	0.1335	0.1768
• LORIA/17+LSI+MT	0.1342	0.1745
Bi-RNN	**0.1359**	**0.1765**
• UGENT-LT3/SCATE-SVM	0.1371	0.1745
Baseline SVM	0.1482	0.1913

Table 1: Proposed approach (Bi-RNN) results and official results for the **scoring variant** of WMT15 Quality Estimation Shared Task at sentence level. A total of 5 tied official winning systems are indicated by a •. Two standard metrics is used: Mean Average Error (MAE) as a primary metric, and Root of Mean Squared Error (RMSE) as a secondary metric (Bojar et al., 2015).

System ID	**DeltaAvg** ↑	Spearman's ρ ↑
• LORIA/17+LSI+MT+FILTRE	6.51	0.36
• LORIA/17+LSI+MT	6.34	0.37
• RTM-DCU/RTM-FS+PLS-SVR	6.34	0.37
• RTM-DCU/RTM-FS-SVR	6.09	0.35
Bi-RNN	**6.08**	**0.33**
Baseline SVM	2.16	0.13

Table 2: Proposed approach (Bi-RNN) results and official results for the **ranking variant** of WMT15 Quality Estimation Shared Task at sentence level. A total of 4 tied official winning systems are indicated by a •. DeltaAvg metric is used as a primary metric (Bojar et al., 2015).

Table 1 and 2 present the results of the proposed approach (Bi-RNN) and the official results for the scoring and ranking[11] variants of the WMT15 Quality Estimation Shared Task at sentence level. At both variants of the task, the proposed RNNs approach achieved the performance over the baseline performance. Also our experiments showed that the performance of the proposed RNNs approach is included to the best performance group (at the scoring variant of Table 1) or is close to the best performance group (at the ranking variant of Table 2).

5 Conclusion

This paper proposed a recurrent neural networks approach using quality vectors for estimating the quality of machine translation output at sentence level.

[9] `http://www.statmt.org/wmt15/quality-estimation-task.html`

[10] Stochastic gradient descent (SGD) algorithm with adaptive learning rate (Adadelta) (Zeiler, 2012) is used to train the proposed model.

[11] The ranking variant of the QE task measures how close a proposed ranking of target translations from best to worst is to the true ranking.

This approach does not require manual effort for finding the best relevant features which the previous QE research has mainly focused on.

To make quality vectors we used an alternative prediction method based on large-scale parallel corpora, because the QE training data were not enough. By extending the prediction method to use bi-directional RNN architecture both on source and target sentence, we were able to fully utilize the bi-directional quality information from source and target sentence for quality estimation.

The proposed RNNs approach achieved a performance comparable to the existing state-of-the-art models at sentence-level QE. Our experiments have showed that RNNs approach is a meaningful step for QE research. Applying RNNs approach to word-level QE and studying other ways to make quality vectors better are remained for the future study.

Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the "ICT Consilience Creative Program" (IITP-2015-R0346-15-1007) supervised by the IITP (Institute for Information & communications Technology Promotion)

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

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.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal, September. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Jesús González-Rubio, J Ramón Navarro-Cerdán, and Francisco Casacuberta. 2013. Dimensionality reduction methods for machine translation quality estimation. *Machine translation*, 27(3-4):281–301.

Ian Goodfellow, Aaron Courville, and Yoshua Bengio. 2015. Deep learning. Book in preparation for MIT Press.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

David Langlois. 2015. Loria system for the wmt15 quality estimation shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 323–329, Lisbon, Portugal, September. Association for Computational Linguistics.

Kashif Shah, Trevor Cohn, and Lucia Specia. 2015. A bayesian non-linear method for feature selection in machine translation quality estimation. *Machine Translation*, pages 1–25.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *13th Conference of the European Association for Machine Translation*, pages 28–37.

Lucia Specia, Kashif Shah, José GC De Souza, and Trevor Cohn. 2013. Quest-a translation quality estimation framework. In *ACL (Conference System Demonstrations)*, pages 79–84. Citeseer.

Matthew D Zeiler. 2012. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.