Translation Quality Estimation using Recurrent Neural Network

Raj Nath Patel
CDAC Mumbai, India
rajnathp@cdac.in

Sasikumar M
CDAC Mumbai, India
sasi@cdac.in

Abstract

This paper describes our submission to the shared task on word/phrase level Quality Estimation (QE) in the First Conference on Statistical Machine Translation (WMT16). The objective of the shared task was to predict if the given word/phrase is a correct/incorrect (OK/BAD) translation in the given sentence. In this paper, we propose a novel approach for word level Quality Estimation using Recurrent Neural Network Language Model (RNN-LM) architecture. RNN-LMs have been found very effective in different Natural Language Processing (NLP) applications. RNN-LM is mainly used for vector space language modeling for different NLP problems. For this task, we modify the architecture of RNN-LM. The modified system predicts a label (OK/BAD) in the slot rather than predicting the word. The input to the system is a word sequence, similar to the standard RNN-LM. The approach is language independent and requires only the translated text for QE. To estimate the phrase level quality, we use the output of the word level QE system.

1 Introduction

Quality estimation is the process to predict the quality of translation without any reference translation (Blatz et al., 2004, Specia et al., 2009). Whereas, Machine Translation (MT) system evaluation does require references (human translation). QE could be done at word, phrase, sentence or document level. This paper describes the submission to the shared task on word and phrase level QE (Task 2) for English-German (en-de) MT. The shared task has the trace of last five years' research in the field of QE (Callision-Burch et al., 2012; Bojar et al., 2013; Bojar et al., 2014; Bojar et al., 2015).

In recent years, RNN-LM has demonstrated exceptional performance in a variety of NLP applications (Mikolov et al., 2010; Mikolov et al., 2013a; Mikolov et al., 2013b; Socher et al., 2013a; Socher et al., 2013b). The RNN-LM represents each word as high-dimensional real-valued vectors, like the other continuous space language models such as feed forward neural network language models (Schwenk and Gauvain, 2002; Bengio et al., 2003; Morin and Bengio, 2005; Schwenk, 2007) and Hierarchical Log-Bi-linear language models (Minh and Hinton, 2009).

In this paper, we have used a modified version of RNN-LM, which accepts the word sequence (context window) as input and predicts label at the output for the middle word. For example, let us consider the following input/output sample:

English (MT input): Layer effects are retained by default .

German (MT output): " Effekte sind standardmig beibehalten .

German (Post-edited): Ebeneneffekte werden standardmig beibehalten .

Tags: BAD BAD BAD OK OK OK

Now if we have to predict the output tag (BAD) for the word "sind" in the MT output, our input sequence to the RNN-LM will be "Effekte sind standardmig" (if context window size is 3). Whereas, for standard RNN-LM model, "Effekte standardmig" would be the input to the network with "sind" as the output. We add padding at the start and end of the sentence according to the context window. The detailed description of the model and its implementation is given in section 3.

We have used the data provided by the or-

Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers, pages 819–824,
Berlin, Germany, August 11-12, 2016.

ganizers for the shared task on quality estimation (2016) which includes: (i) source sentence (ii) translated output (word/phrase level) (iii) word/phrase level tagging (OK/BAD) (iv) post edited translation (v) 22 baseline features (vi) word alignment. The goal of the task is to predict whether the given word/phrase is a correct/incorrect (OK/BAD) translation in the given sentence.

The remainder of the paper is organised as follows. Section 2 describes the related work. Section 3 presents RNN models we use, and its implementation. In section 4, we discuss the data distribution, our approaches, and results. Discussion of our methodology and different models is covered in section 5 followed by concluding remarks in section 6.

2 Related Work

For word level QE, supervised classification techniques are being used widely. Most of these approaches require manually designed features (Bojar et al., 2014), similar to the feature set provided by the organizers.

Logacheva et al. (2015) modeled the word level QE using the CRF++ tool with data selection and data bootstrapping in which data selection filters out the sentences having the smallest proportion of erroneous tokens and are assumed to be less useful for the task. The bootstrapping technique creates additional data instances and boosts the importance of BAD labels occurring in the training data. Shang et al. (2015) tried to solve the problem of label imbalance with creating sub-labels like OK_B (begin), OK_I (intermediate), OK_E (end). Shah et al. (2015) have used word embedding as an additional feature (+25 features) with SVM classifier. Bilingual Deep Neural Network (DNN) based model for word level QE was proposed by Kreutzer et al. (2015), in which word embedding was pre-trained and fine-tuned with other parameters of the network using stochastic gradient descent. de Souza et al. (2014) have used Bidirectional LSTM as a classifier for word level QE.

The architecture of RNN-LM has been used for Natural Language Understanding (NLU) (Yao et al., 2013; Yao et al., 2014) earlier. Our approach is quite similar to the Kreutzer et al. (2015), but we are using RNN instead of DNN. We have also tried to address the problem of label-imbalance, introducing sub-labels as suggested by Shang et al. (2015).

3 RNN Models for QE

For this task, we exploited RNN's extensions, Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014). LSTM and GRU have shown to perform better at modeling the long-range dependencies in the data than the simple RNN. Simple RNN also suffers from the problem of exploding and vanishing gradient (Bengio et al., 1994). LSTM and GRU tackle this problem by introducing a gating mechanism. LSTM includes input, output and forget gates with a memory cell, whereas GRU has reset and update gates only (no memory cell). The detailed description of each model is given in the following subsections.

3.1 LSTM

Different researchers use slightly different LSTM variants (Graves, 2013; Yao et al., 2014; Jozefowicz et al., 2015). We implemented the version of LSTM described by the following set of equations:

$$
\begin{aligned}
i_t &= sigm(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
o_t &= sigm(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
f_t &= sigm(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
j_t &= tanh(W_{xj}x_t + W_{hj}h_{t-1} + b_j) \\
c_t &= c_{t-1} \odot f_t + i_t \odot j_t \\
h_t &= tanh(c_t) \odot o_t
\end{aligned}
$$

where $sigm$ is the logistic sigmoid function and $tanh$ is the hyperbolic tangent function to add non linearity in the network. $\odot$ is the element-wise multiplication of vectors. i, o, f are $input$, $output$, $forget$ gates respectively, j is the new memory content whereas c is the updated memory content. In these equations, W_* are the weight matrices and b_* are the bias vectors.

3.2 Deep LSTM

In this paper, we have used deep LSTM with two layers. Deep LSTM is created by stacking multiple LSTMs on the top of each other. We feed the output of the lower LSTM as the input to the upper LSTM. For example, if h_t is the output of the lower LSTM, we apply a matrix transform to form the input x_t for the upper LSTM. The matrix transformation allows having two consecutive LSTM layers of different sizes.

3.3 GRU

GRU is an architecture, which is quite similar to the LSTM. Chung et al. (2014) found that GRU outperforms LSTM on a suit of tasks. GRU is defined by the following set of equations:

$$r_t = sigm(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$
$$z_t = sigm(W_{xz}x_t + W_{hz}h_{t-1} + b_z)$$
$$\widetilde{h}_t = tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h)$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h}_t$$

In the above equations, W_* are the weight matrices and b_* are the bias vectors. r and z are known as the reset and update gate respectively. GRU does not use any separate memory cell as used in LSTM. However, gated mechanism controls the flow of information in the unit.

3.4 Implementation Details

We implemented all the models (LSTM, deep LSTM and GRU) with [1]THEANO framework (Bergstra et al., 2010; Bastien et al., 2012) as described above. For all the models in the paper, the size of a hidden layer is 100, the word embedding dimensionality is 100 and the context word window size is 5.

We initialized all the square weight matrices as random orthogonal matrices. All the bias vectors were initialized to zero. Other weight matrices were sampled from a Gaussian distribution with mean 0 and variance 0.01^2.

To update the model parameters, we have used Truncated Back-Propagation-Through-Time (T-BPTT) (Werbos, 1990) with stochastic gradient descent. We fixed the depth of BPTT to 7 for all the models. We used Ada-delta (Zeiler, 2012) to adapt the learning rate of each parameter automatically ($\epsilon = 10^{-6}$ and $\rho = 0.95$). We trained each model for 50 epochs.

4 Experiments and Results

In this section, we describe the experiments carried out for the shared task and present the experimental results.

4.1 Data distribution

We have used the corpus shared by the organizers for our experiments. The split for training/development/testing is detailed in table 1. Test1 split was used for evaluating the different experiments that we have carried out for the shared task. Evaluation scores displayed in the results section are against Test1 only. Organizers provided another set of test data (Test2), against which all the submitted systems were evaluated.

	#Sentences	#Tokens
Train	11000	184697
Dev	1000	17777
Test1	1000	16543
Test2	2000	34477

Table 1: Corpus distribusion.

4.2 Methodology

In the following subsections, we discuss our approaches for word/phrase level quality estimation.

4.2.1 Word Level QE

Our experiments are mainly focused on the word level QE. We have used the output of the word level QE system for the estimation of the phrase level quality.

As mentioned above, we have used the modified RNN-LM architecture for the experiments. Baseline (LSTM) system was developed by training word embedding from scratch with other parameters of the model. In another set of experiments, we have pre-trained the word embedding with $word2vec$ (Mikolov et al., 2013b), and further tuned with the training of the model parameters. For pretraining, we have used an additional corpus (2M sentences approx.) from English-German Europarl data (Koehn, 2005).

For bilingual models, we restructured the source sentence (English) according to the target (German) using word alignment provided by the organizers. For many-to-one mapping in the alignment (English-German), we chose the first alignment only. The 'NULL' token was assigned to the words where were not aligned with any word on the target side. The input of the model is constructed by concatenating context words of source and target. For example, consider the source word sequence $s_1s_2s_3$, and the target word sequence $t_1t_2t_3$, then the input to the network will be $s_1s_2s_3t_1t_2t_3$.

In the training data, the distribution of the labels (OK/BAD) is skewed (OK to BAD ratio is approx. 4:1). To handle the issue, we tried one of the strategies proposed by Shang et al. (2015),

[1] http://deeplearning.net/software/theano/#download

in which we replace 'OK' label with sub-labels to balance the distribution. The sub-labels are OK_B, OK_I, OK_E, depending on the location of the token in the sentence.

4.2.2 Phrase Level QE

For phrase level QE, we have not trained any explicit system. As it was mentioned by the organizers that a phrase is tagged as 'BAD', if any word in the phrase is an incorrect translation. So, We have taken the output of the word level QE system and tagged the phrase as 'BAD', if any word in the phrase boundary is tagged 'BAD'. And other phrases (all words have the OK tag) are simply tagged as 'OK'.

Model/Test	F1 BAD	F1 OK
Baseline (LSTM)	35.60	82.93
LSTM_PT	37.27	83.25
LSTM_PT_SL	36.27	81.38
LSTM_BL	36.18	82.51
LSTM_BL_PT	38.53	83.80
LSTM_BL_PT_SL	39.17	83.20
DeepLSTM	35.86	80.35
DeepLSTM_PT	36.81	82.51
DeepLSTM_PT_SL	36.13	81.32
DeepLSTM_BL	37.41	81.92
DeepLSTM_BL_PT	38.38	81.41
DeepLSTM_BL_PT_SL	37.04	82.40
GRU	37.98	84.29
GRU_PT	39.42	84.81
GRU_PT_SL	**40.46**	**83.09**
GRU_BL	41.56	84.57
GRU_BL_PT	42.46	83.76
GRU_BL_PT_SL	42.92	83.62

Table 2: F1 scores of different experiments for Word level QE. (PT: Pretrain; BL: Bilingual; SL: Sublabels)

4.3 Results

To develop a baseline system for word and phrase level QE, organizers have used the baseline features (22 features) to train a Conditional Random Field (CRF) model with CRFSuite tool. The results of the experiments against Test2 are displayed in table 4 and 5.

We have evaluated our systems using the F1-score. As 'OK' class is dominant in the data and a naive system tagging all the words 'OK' will score high. Hence, F1-score of the 'BAD' class has been used as a primary metric for the system evaluation. We have used the separate set of test and development corpus as shown in table 1. The evaluation of all the experiments against Test1 corpus is displayed in table 2 for word level QE. Results for phrase level QE are shown in table 3.

Model/Test	F1 BAD	F1 OK
Baseline (LSTM)	43.46	75.41
LSTM_PT	45.41	75.67
LSTM_PT_SL	44.92	73.11
LSTM_BL	44.43	74.93
LSTM_BL_PT	45.75	77.17
LSTM_BL_PT_SL	46.96	75.73
DeepLSTM	43.83	71.98
DeepLSTM_PT	44.92	74.17
DeepLSTM_PT_SL	43.85	72.32
DeepLSTM_BL	45.65	73.81
DeepLSTM_BL_PT	46.50	72.68
DeepLSTM_BL_PT_SL	45.63	74.57
GRU	45.70	77.86
GRU_PT	46.49	80.00
GRU_PT_SL	**48.38**	**76.14**
GRU_BL	48.11	77.69
GRU_BL_PT	49.58	76.88
GRU_BL_PT_SL	49.61	77.20

Table 3: F1 scores of different experiments for Phrase level QE.

From the result tables, it is evident that GRU outperforms LSTM as reported by Chung et al. (2014) for this task as well. Pre-training is helpful in all the models. Also, the introduction of sub-labels is able to handle the problem of label-imbalance up to some extent. The results of Bilingual models are better than monolingual models, as reported by Kreutzer et al. (2015).

4.4 Submission to the shared task

We have participated in the Task-2, which includes word and phrase level quality estimation. The submitted system setting was: $GRU + Pretrain + Sublabels$, which is **marked** in the result tables (2 and 3) as well. Table 4 and 5 detail the [2]results of the submission on Test2 corpus. The submission results were provided by the organizers.

	F1 BAD	F1 OK
Baseline (CRF)	36.82	88.00
Submitted system	41.92	84.21

Table 4: Results, word level submission.

	F1 BAD	F1 OK
Baseline (CRF)	40.14	80.01
Submitted system	50.31	75.50

Table 5: Results, phrase level submission.

[2]http://www.quest.dcs.shef.ac.uk/wmt16_files_qe/wmt16_task2_results.pdf

5 Discussion

The approach is language independent and it uses only context words' vector for predicting the tag for a word. In the other words, we check if any word fits (grammatically) in the given slot of words or not. We could use language specific features to enhance the classification accuracy, though. Experiments with bilingual models are similar to the concept of adding more features to any machine learning algorithm. In monolingual models, we use only target (German) words' vector as feature whereas, in bilingual models, we use source (English) words' vector also. A challenge which machine learning practitioners often face is, how to deal with skewed classes in classification problems. The distribution of classes (OK/BAD) is skewed in our case as well. To handle the issue, we tried to balance the distribution of classes by introducing the sub-labels.

LSTM and GRU are quite similar models, except the gating mechanism. It is hard to say which model will perform better in what conditions or in general (Chung et al., 2014). In this paper and in general as well, this restricts us to conduct only the empirical comparison between the LSTM and the GRU units. Deep models generally perform better than the shallow models, which is opposite for this task where LSTM outperforms Deep LSTM. The reason could be the insufficient data for training the deep models.

6 Conclusion and Future Work

We have developed a language independent word/phrase level Quality Estimation system using RNN. We have used RNN-LM architecture, with LSTM, deep LSTM, and GRU. We showed that these models benefit from pretraining and the introduction of sub-labels. Also, models with bilingual features outperform the monolingual models.

We can extend the work for sentence and document level quality estimation. Improving the word level quality estimation with data selection and bootstrapping (Logacheva et al., 2015), more effective ways to handle label-imbalance, training bigger models, using language specific feature, other variations of LSTM architecture etc., are the other possibilities.

References

Frederic Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. 2012. Theano: new features and speed improvements. In *NIPS 2012 deep learning workshop*.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. In *IEEE Transactions on Neural Networks*, pages 157–166.

Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. In *Journal of Machine Learning Reseach*, volume 3.

James Bergstra, Olivier Breuleux, Frederic Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4.

John Blatz, Erin Fitzgerald, George Fostera, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation for Machine Translation. In *COLING 2014*, pages 315–321. Geneva, Switzerland.

Ondrej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 workshop on statistical machine translation. In *WMT13*, pages 1–44. Sofia, Bulgaria.

Ondrej Bojar, Christian Buck, Christian Federman, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ales Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *WMT14*, pages 12–58. Baltimore, MD.

Ondej 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 *WMT15*, pages 1–47. Lisbon, Portugal.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *WMT12*, pages 10–51. Montreal, Canada.

Kyunghyun Cho, Bart Van Merrinboer, 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 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *arXiv:1412.3555 [cs.NE]*.

Jose GC de Souza, U. Politecnica de Valencia, Christian Buck, Marco Turchi, and Matteo Negri. 2014. FBK-UPV-UEdin participation in the WMT14 Quality Estimation shared-task. In *WMT14*, pages 322–328.

Alex Graves. 2013. Generating sequences with recurrent neural networks. In *arXiv:1308.0850 [cs.NE]*.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. In *Neural computation*, pages 1735–1780.

Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2342–2350.

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

Julia Kreutzer, Shigehiko Schamoni, and Stefan Riezler. 2015. QUality Estimation from ScraTCH (QUETCH): Deep Learning for Word-level Translation Quality Estimation. In *WMT15*, pages 316–322. Lisboa, Portugal.

Varvara Logacheva, Chris Hokamp, and Lucia Specia. 2015. Data enhancement and selection strategies for the word-level Quality Estimation. In *WMT15*, pages 330–335. Lisboa, Portugal.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech*, volume 2. Makuhari, Chiba, Japan.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013a. Exploiting Similarities among Languages for Machine Translation. In *CoRR*, pages 1–10.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Andriy Mnih and Geoffrey E. Hinton. 2009. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical Probabilistic Neural Network Language Model. In *Aistats*, volume 5, pages 246–252.

Holger Schwenk and Jean-Luc Gauvain. 2002. Connectionist language modeling for large vocabulary continuous speech recognition. In *ICASSP. IEEE*, volume 1, pages I–765.

Holger Schwenk. 2007. Continuous space language models. In *Computer Speech and Language*, volume 21, pages 492–518.

Kashif Shah, Varvara Logacheva, G. Paetzold, Frederic Blain, Daniel Beck, Fethi Bougares, and Lucia Specia. 2015. SHEF-NN: Translation Quality Estimation with Neural Networks. In *WMT15*, pages 342–347. Lisboa, Portugal.

Liugang Shang, Dongfeng Cai, and Duo Ji. 2015. Strategy-Based Technology for Estimating MT Quality. In *WMT15*, pages 248–352. Lisboa, Portugal.

Richard Socher, John Bauer, Christopher D. Manning, , and Andrew Y. Ng. 2013a. Parsing With Compositional Vector Grammars. In *Proceedings of the ACL 2013*, pages 455–465.

Richard Socher, Alex Perelygin, , and Jy Wu. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the Sentence-Level Quality of Machine Translation Systems. In *EACL 2009*, pages 28–37. Barcelona, Spain.

Paul J. Werbos. 1990. Backpropagation through time: what it does and how to do it. In *IEEE*, volume 78, pages 550–1560.

Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. 2013. Recurrent neural networks for language understanding. In *INTERSPEECH*, pages 2524–2528.

Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014. Spoken language understanding using long short-term memory neural networks. In *Spoken Language Technology Workshop (SLT), IEEE*, pages 189–194.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. In *arXiv:1212.5701 [cs.LG]*.