Predictor-Estimator using Multilevel Task Learning with Stack Propagation for Neural Quality Estimation

Hyun Kim[†] and **Jong-Hyeok Lee**[‡]
[†]Creative IT Engineering, [‡]Computer Science and Engineering,
Pohang University of Science and Technology (POSTECH), Republic of Korea
[†]hkim.postech@gmail.com [‡]jhlee@postech.ac.kr

Seung-Hoon Na
Computer Science and Engineering,
Chonbuk National University, Republic of Korea
nash@jbnu.ac.kr

Abstract

In this paper, we present a two-stage neural quality estimation model that uses multilevel task learning for translation quality estimation (QE) at the sentence, word, and phrase levels. Our approach is based on an end-to-end stacked neural model named *Predictor-Estimator*, which has two stages consisting of a neural word prediction model and neural QE model. To efficiently train the two-stage model, a *stack propagation* method is applied, thereby enabling us to jointly learn the word prediction model and QE model in a single learning mode. In addition, we deploy multilevel task learning with stack propagation, where the training examples available for all QE subtasks (i.e., sentence/word/phrase levels) are used to train a Predictor-Estimator for a specific subtask. All of our submissions to the QE task of WMT17 are ensembles that combine a set of neural models trained under different settings of varying dimensionalities and shuffling training examples, eventually achieving the best performances for all subtasks at the sentence, word, and phrase levels.

1 Introduction

In this paper, we describe the two-stage end-to-end neural models submitted to the Shared Task on Sentence/Word/Phrase-Level Quality Estimation (QE task) at the 2017 Conference on Machine Translation (WMT17). The task aims at estimating quality scores/categories for an unseen translation without a reference translation at various granularities (i.e., sentence/word/phrase levels) (Specia et al., 2013).

Our neural network-based models for sentence/word/phrase-level QE are based on Predictor-Estimator architecture (Kim et al., 2017; Kim and Lee, 2016), which is a two-stage end-to-end neural QE model. In this submission to WMT 2017, our Predictor-Estimator model is further advanced by extensively applying a stack propagation method (Zhang and Weiss, 2016) in order to efficiently train the two-stage model.

The Predictor-Estimator architecture (Kim et al., 2017; Kim and Lee, 2016) is the two-stage neural QE model (Figure 1) consisting of two types of stacked neural models: 1) a neural word prediction model (i.e., word predictor) trained from additional large-scale parallel corpora and 2) a neural QE model (i.e., quality estimator) trained from quality-annotated noisy parallel corpora called *QE data*. The Predictor-Estimator architecture uses word prediction as a pre-task for QE. Kim et al. (2017) showed that word prediction is helpful for improving the QE performance. In the first stage, the word predictor, which is based on a bidirectional and bilingual recurrent neural network (RNN) language model – the modification of the attention-based RNN encoder-decoder (Bahdanau et al., 2015; Cho et al., 2014) – predicts a target word conditioned with unbounded source and target contexts. QE feature vectors (QEFVs) are the approximated knowledge transferred from word prediction to QE. In the second stage, QEFVs are used as inputs to the quality estimator for estimating sentence/word/phrase-level translation quality.

Stack propagation (Zhang and Weiss, 2016) is a learning method for efficient joint learning that enables backpropagation down the stacked models. Zhang and Weiss (2016) applied stack propagation for stacked part-of-speech (POS) tagging and parsing models by alternating between stochastic updates to POS tagging or parsing objectives,

Proceedings of the Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 562–568
Copenhagen, Denmark, September 711, 2017. ©2017 Association for Computational Linguistics

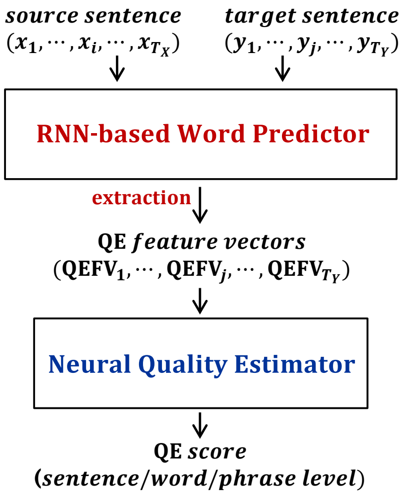
source sentence
$(x_1, \cdots, x_i, \cdots, x_{T_X})$
target sentence
$(y_1, \cdots, y_j, \cdots, y_{T_Y})$
RNN-based Word Predictor
extraction
QE feature vectors
$(\mathrm{QEFV}_1, \cdots, \mathrm{QEFV}_j, \cdots, \mathrm{QEFV}_{T_Y})$
Neural Quality Estimator
QE score
(sentence/word/phrase level)

Figure 1: Two-stage Predictor-Estimator architecture (Kim et al., 2017).

where continuous hidden layer activations of the POS tagger network are used as an input to the parser network.

We applied the Predictor-Estimator architecture to the sentence/word/phrase-level QE task of WMT17. In the original Predictor-Estimator architecture proposed by Kim et al. (2017), the word predictor and quality estimator are trained individually. As a result, the backpropagation in training the quality estimator does not go down for the word predictor network. Because there exists a continuous and differentiable link between the stacked word predictor and quality estimator, we used stack propagation to jointly learn two-stage models in the Predictor-Estimator. Furthermore, we deployed multilevel task learning with stack propagation, where a task-specific Predictor-Estimator is trained by using not only the task-specific training examples but also all other training examples of QE subtasks. Finally, all of our submissions for the QE task of WMT17 were ensembles that combine a set of neural models trained under different settings of varying dimensionalities and shuffled training examples.

2 Improving Predictor-Estimator with Stack Propagation

In this section, we describe the three types of Predictor-Estimators using stack propagation: 1) the base model (*PredictorEstimator*), 2) Predictor-Estimator using stack propagation for a single-level task (*PredictorEstimator + (SingleLevel) Stackprop*), and 3) Predictor-Estimator using multilevel task learning with stack propagation (*PredictorEstimator + MultiLevel Stackprop*).

2.1 Base Model

Our base model is the original Predictor-Estimator, where a word predictor and quality estimator are trained individually. We used the Pre&Post-QEFV/Bi-RNN model, which showed the best performance among the Predictor-Estimator models presented by Kim et al. (2017). The Pre&Post-QEFV/Bi-RNN model is a two-stage model that uses Pre&Post-QEFV extracted from the word predictor and Bi-RNN applied in the quality estimator. Pre&Post-QEFV is the summary representation in the word predictor networks and involves approximating the transferred knowledge from each target word prediction. This consists of the word prediction-based weight-inclusive indirect representation (i.e., Pre-QEFV) and direct hidden state (i.e., Post-QEFV).

2.2 Using Stack Propagation

Because the Predictor-Estimator architecture has a continuous and differentiable link between the stacked word predictor and quality estimator, allowing backpropagation from the quality estimator to the word predictor is a valuable approach. To jointly learn the two-stage models in the Predictor-Estimator, stack propagation is applied by alternating between stochastic updates to word prediction or QE objectives, thus performing backpropagation down from the quality estimator to the word predictor (Figure 2).

2.3 Using Multilevel Task Learning with Stack Propagation

We implemented multilevel task learning with stack propagation that uses the training examples available for all QE subtasks (sentence/word/phrase level) to train a task-specific Predictor-Estimator. There are mutual common parts in the Predictor-Estimator networks for sentence/word/phrase-level QE: 1) all of the word predictor networks and 2) input parts and hidden states of the quality estimator networks, except for the output parts at each level. In multilevel task learning with stack propagation, these common parts of the task-specific Predictor-Estimator networks are trained by using not only task-specific training examples but also all of the other training examples of QE subtasks.

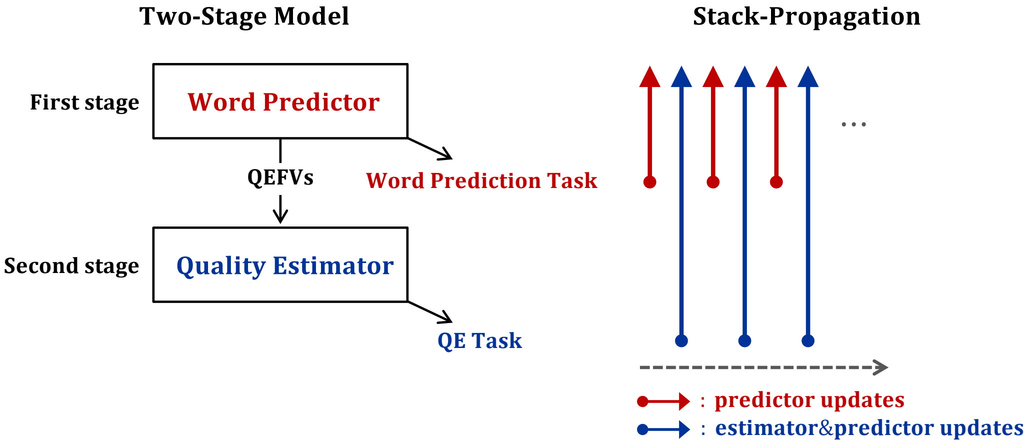

Figure 2: Applied stack propagation (Zhang and Weiss, 2016) to Predictor-Estimator architecture by alternating stochastic updates.

This approach is based on the idea that QE at all levels has a common origin because quality annotations at each level of QE data[1] are obtained by comparing the same post-edited target references with the same target translations to calculate the human-targeted translation edit rate (HTER) (Snover et al., 2006). By using multilevel task learning with stack propagation, mutually beneficial relationships can be learned between each level. We alternate not only between stochastic updates to word prediction or QE objectives but also between stochastic updates to sentence/word/phrase-level QE objectives for jointly learning mutual common parts of the Predictor-Estimator network[2].

3 Experimental Results

3.1 Experimental settings

We evaluated our models for the WMT17 QE task of sentence/word/phrase-level English-German and German-English. To train our two-stage models, we used QE data for the WMT17 QE task (Specia and Logacheva, 2017) and parallel corpora including the Europarl corpus, common crawl corpus, news commentary, rapid corpus of EU press releases for the WMT17 translation task[3], and src-pe (source sentences-their target post-editions) pairs for the WMT17 QE task. All Predictor-Estimator models were initialized with a word predictor and quality estimator that were pre-trained individually.

3.2 Results of the Single Predictor-Estimator Models

For a single Predictor-Estimator model, we used one type of dimensionality settings[4].

Table 1 presents the experimental results for the single Predictor-Estimator models with the English-German QE development set at the sentence, word, and phrase levels. Among the three types of models, the Predictor-Estimator using multilevel task learning with stack propagation consistently exhibited the best performance in all of our runs. Because this was the most sophisticated among our three types of models, we believe that applying more advanced approaches to Predictor-Estimator brings further improvements. The base model, which was the simplest Predictor-Estimator model, exhibited somewhat lower performance than others. The models using stack propagation for sentence/word/phrase-level QE consistently performed better than the base models without stack propagation. This result means

[1] QE data consist of source sentences, target translations (not references), and their target quality annotations for sentence/word/phrase levels.

[2] An original phrase-level Predictor-Estimator and original word-level Predictor-Estimator have different architectures in that the input of the former is phrase-level QEFV, which is the average of its constituent word-level QEFVs. However, in multilevel task learning with stack propagation for phrase-level QE, we use a word-level Predictor-Estimator architecture. In the word-level Predictor-Estimator for phrase-level QE, if any word in the phrase boundary is tagged as 'BAD,' the output of the phrase level has a 'BAD' tag, which exactly corresponds with the purpose of the phrase-level QE.

[3] http://www.statmt.org/wmt17/translation-task.html

[4] The vocabulary size was 70,000, the word embedding dimensionality was 500, the size of the hidden units of the word predictor was 700, and the size of the hidden units of the quality estimator was 100.

Sentence Level	**Pearson's r** ↑	MAE ↓	RMSE ↓	**Spearman's ρ** ↑	DeltaAvg ↑
PredictorEstimator	0.6436	0.1125	0.1582	0.6851	0.1190
+ (SingleLevel) Stackprop	0.6476	0.1122	0.1567	0.6957	0.1209
+ MultiLevel Stackprop	0.6785	0.1047	0.1502	0.7267	0.1234

Word Level	F_1**-mult** ↑	F_1-BAD ↑	F_1-OK ↑
PredictorEstimator	0.5104	0.5747	0.8881
+ (SingleLevel) Stackprop	0.5335	0.5906	0.9034
+ MultiLevel Stackprop	0.5374	0.6018	0.8930

Phrase Level	F_1**-mult** ↑	F_1-BAD ↑	F_1-OK ↑
PredictorEstimator	0.5262	0.6367	0.8264
+ (SingleLevel) Stackprop	0.5631	0.6674	0.8438
+ MultiLevel Stackprop	0.5664	0.6697	0.8457

Table 1: Results of the single Predictor-Estimator models on the WMT17 En-De dev set.

Sentence Level	**Pearson's r** ↑	MAE ↓	RMSE ↓	**Spearman's ρ** ↑	DeltaAvg ↑
PredictorEstimator	0.6375	0.1094	0.1480	0.6665	0.1138
+ (SingleLevel) Stackprop	0.6377	0.1092	0.1473	0.6698	0.1149
+ MultiLevel Stackprop	0.6599	0.1057	0.1450	0.6914	0.1188

Word Level	F_1**-mult** ↑	F_1-BAD ↑	F_1-OK ↑
PredictorEstimator	0.5086	0.5768	0.8818
+ (SingleLevel) Stackprop	0.5203	0.5898	0.8822
+ MultiLevel Stackprop	0.5287	0.5951	0.8883

Phrase Level	F_1**-mult** ↑	F_1-BAD ↑	F_1-OK ↑
PredictorEstimator	0.5116	0.6227	0.8216
+ (SingleLevel) Stackprop	0.5512	0.6522	0.8452
+ MultiLevel Stackprop	0.5527	0.6523	0.8473

Table 2: Results of the single Predictor-Estimator models on the WMT17 En-De test set.

that stack propagation is advantageous for efficient joint learning. The use of multilevel task learning with stack propagation for sentence-level QE significantly improved the QE performance. The use of single-level stack propagation for word/phrase-level QE also significantly improved the QE performance.

Tables 2-3 present the experimental results of the single Predictor-Estimator models for the English-German and German-English QE test set at the different levels.

3.3 Results of Ensembles of Multiple Instances

To develop ensemble-based submissions for the WMT17 QE task, we used two types of single models: the simplest (base model) and most sophisticated (Predictor-Estimator using multilevel task learning with stack propagation).

Martins et al. (2016) combined 15 instances of neural models to make ensembles; they used three types of neural models and trained five instances for each type by using different data shuffles.

In our experiments, we made ensembles of multiple instances trained under different settings of varying dimensionalities and shuffled training examples for the two selected models (i.e., the simplest and the most sophisticated single models). We averaged the predicted scores from each instance for producing the ensemble results. The ensembles for the simplest single model were made by averaging 15 predictions from each single model with five types of dimensionality settings[5] to produce three trained instances with the different shuffling training examples, called *PredictorEstimator-Ensemble*[6].

[5] 1) The vocabulary size was 70,000 words, the word embedding dimensionality was 500, the size of the hidden units of the word predictor was 700, and the size of the hidden units of the quality estimator was 100. 2) The vocabulary size was 70,000 words, the word embedding dimensionality was 500, the size of the hidden units of the word predictor was 700, and the size of the hidden units of the quality estimator was 150. 3) The vocabulary size was 100,000 words, the word embedding dimensionality was 700, the size of the hidden units of the word predictor was 1000, and the size of the hidden units of the quality estimator was 100. 4) The vocabulary size was 100,000 words, the word embedding dimensionality was 700, the size of the hidden units of the word predictor was 1000, and the size of the hidden units of the quality estimator was 150. 5) The vocabulary size was 100,000 words, the word embedding dimensionality was 700, the size of the hidden units of the word predictor was 1000, and the size of the hidden units of the quality estimator was 200.

[6] In the submissions for WMT17 QE task,

Sentence Level	**Pearson's** r ↑	MAE ↓	RMSE ↓	**Spearman's** ρ ↑	DeltaAvg ↑
PredictorEstimator	0.6826	0.0987	0.1428	0.6065	0.1010
+ (SingleLevel) Stackprop	0.6888	0.0977	0.1458	0.6202	0.1026
+ MultiLevel Stackprop	0.6985	0.0952	0.1461	0.6408	0.1039

Word Level	F_1-**mult** ↑	F_1-BAD ↑	F_1-OK ↑
PredictorEstimator	0.4864	0.5259	0.9249
+ (SingleLevel) Stackprop	0.5008	0.5361	0.9342
+ MultiLevel Stackprop	0.5051	0.5411	0.9334

Phrase Level	F_1-**mult** ↑	F_1-BAD ↑	F_1-OK ↑
PredictorEstimator	0.5069	0.5674	0.8934
+ (SingleLevel) Stackprop	0.5143	0.5671	0.9068
+ MultiLevel Stackprop	0.5246	0.5829	0.8999

Table 3: Results of the single Predictor-Estimator models on the WMT17 De-En test set.

Sentence Level (Scoring Variant)	**Pearson's** r ↑	MAE ↓	RMSE ↓	Rank
PredictorEstimator-Ensemble	0.6731	0.1067	0.1412	2
PredictorEstimator-MultiLevel-Ensemble	0.6891	0.1016	0.1390	
PredictorEstimator-Combined-MultiLevel-Ensemble	0.6954	0.1019	0.1371	1
BASELINE	0.397	0.136	0.175	

Sentence Level (Ranking Variant)	**Spearman's** ρ ↑	DeltaAvg ↑	Rank
PredictorEstimator-Ensemble	0.7029	0.1198	2
PredictorEstimator-MultiLevel-Ensemble	0.7194	0.1221	
PredictorEstimator-Combined-MultiLevel-Ensemble	0.7253	0.1232	1
BASELINE	0.425	0.0745	

Word Level	F_1-**mult** ↑	F_1-BAD ↑	F_1-OK ↑	Rank
PredictorEstimator-Ensemble	0.5429	0.6069	0.8945	5
PredictorEstimator-MultiLevel-Ensemble	0.5602	0.6210	0.9021	
PredictorEstimator-Combined-MultiLevel-Ensemble	0.5679	0.6283	0.9039	1
BASELINE	0.361	0.407	0.886	

Phrase Level	F_1-**mult** ↑	F_1-BAD ↑	F_1-OK ↑	Rank
PredictorEstimator-Ensemble	0.5492	0.6518	0.8426	2
PredictorEstimator-MultiLevel-Ensemble	0.5808	0.6728	0.8633	
PredictorEstimator-Combined-MultiLevel-Ensemble	0.5859	0.6787	0.8633	1
BASELINE	0.327	0.402	0.814	

Table 4: Results of ensembles of multi-instance Predictor-Estimator models on the WMT17 En-De test set.

Sentence Level (Scoring Variant)	**Pearson's** r ↑	MAE ↓	RMSE ↓	Rank
PredictorEstimator-Ensemble	0.7146	0.0942	0.1359	2
PredictorEstimator-MultiLevel-Ensemble	0.7170	0.0907	0.1359	
PredictorEstimator-Combined-MultiLevel-Ensemble	0.7280	0.0911	0.1332	1
BASELINE	0.441	0.128	0.175	

Sentence Level (Ranking Variant)	**Spearman's** ρ ↑	DeltaAvg ↑	Rank
PredictorEstimator-Ensemble	0.6327	0.1044	2
PredictorEstimator-MultiLevel-Ensemble	0.6550	0.1061	
PredictorEstimator-Combined-MultiLevel-Ensemble	0.6542	0.1064	1
BASELINE	0.45	0.0681	

Word Level	F_1-**mult** ↑	F_1-BAD ↑	F_1-OK ↑	Rank
PredictorEstimator-Ensemble	0.5160	0.5516	0.9356	3
PredictorEstimator-MultiLevel-Ensemble	0.5271	0.5609	0.9398	
PredictorEstimator-Combined-MultiLevel-Ensemble	0.5347	0.5687	0.9402	1
BASELINE	0.342	0.365	0.939	

Phrase Level	F_1-**mult** ↑	F_1-BAD ↑	F_1-OK ↑	Rank
PredictorEstimator-Ensemble	0.5428	0.5990	0.9062	2
PredictorEstimator-MultiLevel-Ensemble	0.5490	0.6032	0.9101	
PredictorEstimator-Combined-MultiLevel-Ensemble	0.5611	0.6150	0.9122	1
BASELINE	0.360	0.397	0.907	

Table 5: Results of ensembles of multi-instance Predictor-Estimator models on WMT17 De-En test set.

Ensembles for the most sophisticated single model were made by averaging 15 predictions yielded from each single model with three types of dimensionality settings[7] to produce five trained instances with different shuffling training examples, called *PredictorEstimator-MultiLevel-Ensemble*. We also created an ensemble that combines both *PredictorEstimator-Ensemble* and *PredictorEstimator-MultiLevel-Ensemble*, called *PredictorEstimator-Combined-MultiLevel-Ensemble*.

Tables 4-5 present the experimental results for the ensembles of multi-instance Predictor-Estimator models with the English-German/German-English test set for sentence-/word-/phrase-level QE[8]. In all of our runs, *PredictorEstimator-Combined-MultiLevel-Ensemble* exhibited the best performance and was ranked first for all subtasks at the different levels for the WMT17 QE task.

4 Conclusion

We presented a two-stage end-to-end neural QE model that uses multilevel task learning with stack propagation for sentence/word/phrase-level QE. We used the Predictor-Estimator architecture (Kim et al., 2017; Kim and Lee, 2016) for sentence/word/phrase-level QE. We applied stack propagation (Zhang and Weiss, 2016) to the Predictor-Estimator architecture for efficient joint learning. Finally, we deployed multilevel task learning with stack propagation to use the training examples available for all QE subtasks to train a task-specific Predictor-Estimator. We developed ensembles by combining a set of neural models trained under different settings of varying dimensionalities and shuffling training examples. Our ensemble-based submissions achieved the best performances for all subtasks at the various levels for the WMT17 QE task.

Acknowledgments

This work was partly supported by the ICT R&D Program of MSIP/IITP and ICT Consilience Creative Program of MSIP/IITP.

References

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

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)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734. http://www.aclweb.org/anthology/D14-1179.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. In *ACM Trans. Asian Low-Resour. Lang. Inf. Process (in press)*.

Hyun Kim and Jong-Hyeok Lee. 2016. A recurrent neural networks approach for estimating the quality of machine translation output. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 494–498. http://www.aclweb.org/anthology/N16-1059.

André F. T. Martins, Ramón Astudillo, Chris Hokamp, and Fabio Kepler. 2016. Unbabel's participation in the wmt16 word-level translation quality estimation shared task. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 806–811. http://www.aclweb.org/anthology/W/W16/W16-2387.

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 and Varvara Logacheva. 2017. WMT17 quality estimation shared task training and development data. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University. http://hdl.handle.net/11372/LRT-1974.

PredictorEstimator-Ensemble was denoted as *PredictorEstimator-SingleLevel-Ensemble*.

[7] 1) The vocabulary size was 70,000 words, the word embedding dimensionality was 500, the size of the hidden units of the word predictor was 700, and the size of the hidden units of the quality estimator was 100. 2) The vocabulary size was 70,000 words, the word embedding dimensionality was 500, the size of the hidden units of the word predictor was 700, and the size of the hidden units of the quality estimator was 150. 3) The vocabulary size was 70,000 words, the word embedding dimensionality was 500, the size of the hidden units of the word predictor was 700, and the size of the hidden units of the quality estimator was 200.

[8] *PredictorEstimator-Combined-MultiLevel-Ensemble* and *PredictorEstimator-Ensemble* were our two submissions for the WMT17 QE task.

Lucia Specia, Kashif Shah, Jose G.C. de 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*. Association for Computational Linguistics, Sofia, Bulgaria, pages 79–84. http://www.aclweb.org/anthology/P13-4014.

Yuan Zhang and David Weiss. 2016. Stack-propagation: Improved representation learning for syntax. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1557–1566. http://www.aclweb.org/anthology/P16-1147.