Human Language Technologies – The Baltic Perspective
K. Muischnek and K. Müürisep (Eds.)

doi:10.3233/978-1-61499-912-6-175

Low-Resource Translation Quality Estimation for Estonian

Elizaveta YANKOVSKAYA [1], Mark FISHEL
University of Tartu, Institute of Computer Science

Abstract. Quality estimation is an essential step in applying machine translation systems in practice, however state-of-the-art approaches require manual post-edits and other expensive resources. We introduce an approach to quality estimation that uses the attention weights of a neural machine translation system and can be applied to a translation produced by any machine translation system; a lighter version of the approach does not even require any post-edits. Our experiments with German-Estonian and English-Estonian translations show that its performance matches the state-of-the-art baseline.

Keywords. quality estimation, machine translation, attention weights

1. Introduction

Over the past few years, the quality of machine translation has grown significantly [1,2]. At the same time, translation quality between different translations by the same system can vary greatly and thus automatic quality estimation is necessary to detect unreliable translations. One of the main drawbacks of the current state-of-the-art in quality estimation [3,4] is that it requires lots of manual post-edits to train as well as computing the input features based on additional resources like language models, n-gram frequencies or alignment probability files.

In this work we propose a low-resource quality estimation method that does not depend on post-edits and only uses the internal parameters of neural machine translation (NMT) systems. We show that the described method works even if the "parameters" did not come from the system that produced the translation or if the translation was done by statistical machine translation system. Experimental evaluation is done on English-Estonian and German-Estonian and the obtained results are comparable in performance to the baseline method [3] while requiring fewer resources.

The work described in this paper is part of a collaboration project between Grata OÜ[2], a translation company, and the University of Tartu as the technical partner. All in-domain texts come from the translation agency, and the resulting translation and quality estimation systems are used by the client in practice.

[1]Corresponding Author: elizaveta.yankovskaya@ut.ee
[2]http://grata.ee

2. Related Work

One of the widespread approaches for quality estimation is to apply the QuEst framework [3]. It consists of two modules: a feature extractor and a machine learning module. The extractor allows to get different features like a number of tokens of input and translated sentences, a language model probability of input and translated segments, a ratio of a number of tokens of input and translated sentences. The obtained features are used as input in the machine learning module that trains classification and regression models and predicts quality estimation scores.

There are two models that showed the best performance on WMT shared task on quality estimation[3] in 2017 [5]. Both models use deep learning methods to estimate the translation's quality.

A neural model stacked into a linear sequential model was proposed in [4,6]. The main features of the linear model are features based on the target word and its aligned source word and their contexts. The neural model gets as input the source and target sentences, their word level alignments and corresponding part-of-speech tags and produces the binary output (*OK*/*BAD*) for each word that is used as an additional feature to the linear sequential model. The result of this stacked architecture is the sequence of binary labels (*OK*/*BAD*). For sequence level quality estimation authors use the fraction of *BAD* labels to compute the HTER (the normalized edit distance) [7]. Besides that, they compute HTER by using a quality estimation model based on an automatic post-editing model (APE) [8]. To get the final predicted HTER score, they compute the average of two HTER scores obtained by using their stacked and APE-based models.

The second model [9] uses two-step neural architecture that called a predictor-estimator architecture [10,11]. During the first step, the neural word prediction model trained on parallel corpora predicts a word in the target sentence. Also, authors extract from the first model quality estimation feature vectors that are inner parameters of the neural model. The obtained vectors are used as inputs of a regression layer of the neural quality estimation model within the second step of the whole model. Authors improved the original predictor-estimator architecture [10] adding stack propagation [12] to jointly learn a two-step model in the predictor-estimator.

Unlike the approaches [3,4,6], our method requires only metrics produced by an NMT system. In contrast to the model of [9] that is based on the predictor-estimator architecture our model has a simpler architecture and in case of NMT systems with the attention mechanism requires only a regression or classification step.

3. Attention Weights for Quality Estimation

3.1. Attention Weights

Based on the encoder-decoder machine translation approach [13,14], a new architecture with an attention mechanism was proposed that allows to align and translate simultaneously [15]. In the new architecture, the decoder uses all hidden states of the encoder instead of the last hidden state and focuses on a particular part of the source sentence when generating each output token during decoding. So, the output of translation systems de-

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

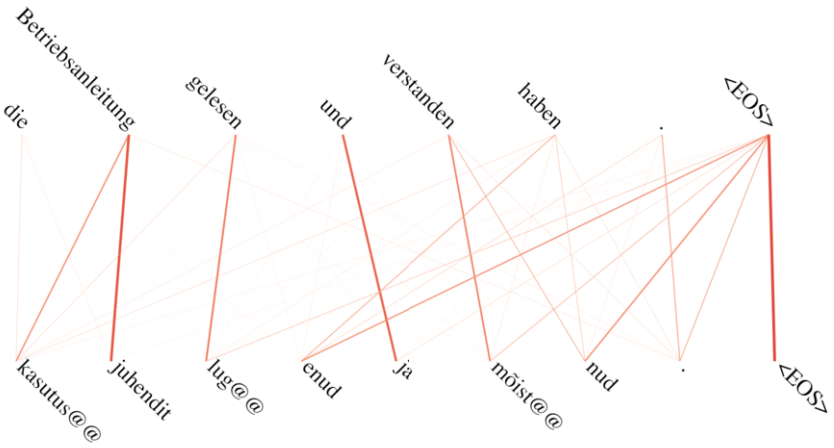

Figure 1. Attention alignment visualization of a good translation [16].

pends on the decoder's hidden state and the attention output. The latter is a probability distribution of the dot product of attention scores and the encoder's hidden state at this time step. The attention weight α_{ij} between the input token j and the output token i is computed as

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_k exp(e_{ik})}$$

where e_{ij} is an attention score shows how well hidden states of encoder and decoder match.

3.2. Attention Weights for Quality Estimation

A visualization of the attention weights of a well translated sentence is shown in Figure 1. It can be seen that the attention weights depict the strength of connection between input and output tokens. [17] have shown that these attention weights can be used for confidence estimation, but only apply it to the case where the attention weights were computed together with the translations. We expand this approach to estimate translation quality of any translations without access to the system that produced them by performing force-decoding with an external neural machine translation (NMT) system.

We have taken the attention-based confidence metrics proposed in [17]:

- **Coverage Deviation Penalty (CDP)** penalizes the sum of attentions per input token, so tokens with less or too much attention get a lower score.

$$CDP = -\frac{1}{I}\sum_j \log\left(1 + (1 - \sum_i \alpha_{ji})^2\right),$$

where I is the length of the input sentence.

- **Absentmindedness Penalty (APin and APout)** computes the dispersion via the entropy of the attention distribution of input and output tokens.

$$AP_{in} = -\frac{1}{I}\sum_{j}\sum_{i}\alpha_{ij}\cdot\log\alpha_{ij}$$

$$AP_{out} = -\frac{1}{I}\sum_{i}\sum_{j}\alpha_{ji}\cdot\log\alpha_{ji}$$

- **Total** is a sum of all metrics described above.

$$Total = CDP + AP_{in} + AP_{out}$$

In addition to the metrics listed above, we also compute the ratio between input and output absentmindedness penalties:

$$AP_{ratio} = \frac{AP_{in}}{AP_{out}}$$

4. Force-decoded Attention Weights

Rikters and Fishel [17] used the attention weights produced by the the system that produced the translation.

We expand their approach by replacing decoding with computing the probability of any translation pair under a separately trained NMT system (force-decoding) and using the resulting attention weights to estimate the translation quality of that translation pair. So, one might say that the attention weights produced by the same system that produced the translation are glass-box features whereas the force-decoded attention weights are black-box features.

As a result, we can use this method to get the attention weights and estimate the translation quality of any translations without access to the system that produced them and regardless of the approach or whether it had an attention mechanism.

5. Experiments and Results

The intended end-application of the quality estimation system in our work is filtering out the worst translations, which are slow to post-edit and are faster to translate manually from the start. We thus classify translated sentences as “acceptable” or “unacceptable”[4]. By “acceptable” sentences we mean sentences in which we need to make no more than a certain number of edits to convert them into post-edited output. Possible edits include deletion, insertion, substitution and shifts of words. To count the number of edits, we use denormalized HTER.

Table 1. Example of normalized and denormalized HTER values. Notations: S – source , MT – machine translation output, PE – post-edited output. Translation: "Kraana" translates as a hoisting machine; "kurg" – a long-necked bird; "mulle meeldivad" – " I like", "konnad" – frogs, "koerad ja kassid" – "dogs and cats"

		normalized HTER	denormalized HTER
1	S: Crane MT: Kraana PE: Kurg	1	1
2	S: I like dogs and cats MT: Mulle meeldivad konnad PE: Mulle meeldivad koerad ja kassid	0.6	3

5.1. Denormalized HTER

After the first experiments we discovered that normalized HTER[5] [7] does not reflect the required post-editing effort; as Table 1 shows we need to replace only one word in the first translation to correct the output, whereas in the second translation we need to make three edits. However, according to HTER values (the lower the value, the better the sentence), the second translated sentence takes less effort to convert it to a correct post-edited output compared to the first sentence.

To strengthen this point of view, we have computed the Pearson correlation coefficient between the time needed for post-editing and normalized/denormalized HTER. We have taken data provided by organizers of WMT18[6] for German-English and English-German language pairs. For German-English language pair we have got the correlation 0.254 for normalized HTER and 0.4 for denormalized HTER; for English-German language pair the correlation coefficient is 0.284 for normalized HTER and 0.390 for denormalized HTER. All obtained coefficients show a weak correlation but the correlation between time and denormalized HTER is stronger than between time and normalized HTER.

To avoid the ambiguity described above, we used the denormalized HTER values, obtained by computing HTER but not normalizing them with the sentence length.

5.2. Experimental Details

The aim of the experiments was to compare the usage of force-decoded weights to the internal parameters of the system that produced the translation, as well as compare attention-based quality estimation to the baseline of QuEst.

After several initial tests, we have chosen Random Forest as the classification algorithm. We ran experiments with different sets of metrics as classifier's features:

- features based on the internal/force-decoded attention weights: CDP, AP_{in}, AP_{out}, $total$, AP_{ratio};
- QuEst features: a standard set of 17 black-box features [3];
- the combination of features based on the attention weights and features based on QuEst.

[4]The more usual approach of doing regression of the output score was not possible due to the lack of data

[5]Number of edits divided by number of tokens of post-edited output.

[6]`http://www.statmt.org/wmt18/quality-estimation-task.html`

To evaluate the performance of the classification algorithm we use recall, precision, F1 score and the Matthews correlation coefficient. The Matthews correlation coefficient (MCC) takes into account true and false positives and negatives and can be used for unbalanced data. It is computed as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

5.3. Data

We used technical texts written in German and English, their translation into Estonian and their manual post-edits.

Translations were done with an NMT system based on Nematus [18]. The system was trained on general corpora and tuned on an in-domain corpus. In all our experiments, we used two RNN (recurrent neural network) layers of size 1024, a word vector of size 512, and adam [19] as optimizer with a learning rate of 0.0001, and a batch sizes of 32. Post-edits were done by our industrial partner. As a result, we had 5444 sentences for German-Estonian and 4541 sentences for English-Estonian.

For German-Estonian we used 4835 sentences for training and 609 sentences as the test set. We set a threshold for seven edits since it results in 75% of the training set containing "acceptable" sentences. For English-Estonian pair we had a training and test sets 4168 and 373 sentences respectively with the acceptable threshold of six edits.

5.4. Results and Discussion

Tables 2 and 3 show the results of the experiments for German-Estonian and English-Estonian language pairs. It can be seen that results for metrics based on the internal and force-decoded attention weights are quite similar, which shows that in our case access to internal parameters of the NMT system that produced the translations is not required. We got the best results by using the combination of features. Still, values obtained by using the attention weights' features are only slightly worse than results obtained by using QuEst features. It means that we can use the proposed low-resource quality estimation method instead of the high-resource QuEst.

Table 2. Results of experiments for German-Estonian language pair: recall, precision, F1 score and the Matthews correlation coefficient (MCC). In this table we show the results for features based on internal weights ("Int") and for features based on force-decoded weights ("FD").

	Recall		Precision		F1		MCC	
	Int	FD	Int	FD	Int	FD	Int	FD
weights	0.906	0.874	0.874	0.892	0.89	0.883	0.593	0.598
QuEst	0.927		0.862		0.893		0.589	
weights+ QuEst	0.908	0.888	0.880	0.894	0.894	0.891	0.611	0.617

In addition to the classification, we computed the correlation between the *total* confidence metric and the denormalized HTER for the internal and force-decoded attention weights. The aim is to see if the attention metrics can be used without training a classifier/regression model and without post-edits for training.

Table 3. Results of experiments for English-Estonian language pair: recall, precision, F1 score and the Matthews correlation coefficient (MCC). In this table we show the results for features based on internal weights ("Int") and for features based on force-decoded weights ("FD").

	Recall		Precision		F1		MCC	
	Int	FD	Int	FD	Int	FD	Int	FD
weights	0.760	0.858	0.901	0.767	0.824	0.810	0.659	0.556
QuEst	0.872		0.864		0.868		0.708	
weights+ QuEst	0.912	0.799	0.873	0.953	0.892	0.869	0.756	0.751

For German-Estonian language pair the resulting Pearson correlation coefficient is -0.668 for the internal weights and -0.467 for force-decoded weights; for English-Estonian the correlation coefficient is -0.689 for the internal weights and -0.412 for force-decoded weights. While the correlation coefficient for the internal and force-decoded weights show a moderate correlation, internal weights produce a better result; still, an approximate output can be obtained by using the attention weights only.

6. Conclusions

We described a method that estimates a quality of translated sentences based on the attention weights of an NMT system. The proposed method can be used not only to estimate the quality of sentences produced by attention NMT system but also to evaluate the quality of translations produced by any MT systems.

We demonstrated that our method works well for classification tasks, it remains to be tested how well it works as a feature of a regression model.

Our experimental results showed that the proposed method gives comparable results with the baseline method without additional effort. Using our method directly without a classifier is possible, though, in this case, force-decoded weights have decreased performance.

Acknowledgments

This work was supported by the Estonian Research Council grant no. 1226. The collaboration project between Grata OÜ and the University of Tartu was partially supported by Enterprise Estonia. The authors are thankful to both funding agencies as well as to the stakeholder for supporting this research.

References

[1] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016.

[2] H. Hassan, A. Aue, C. Chen, V. Chowdhary, J. Clark, C. Federmann, X. Huang, M. Junczys-Dowmunt, W. Lewis, M. Li, S. Liu, T. Liu, R. Luo, A. Menezes, T. Qin, F. Seide, X. Tan, F. Tian, L. Wu, S. Wu, Y. Xia, D. Zhang, Z. Zhang, and M. Zhou, "Achieving human parity on automatic chinese to english news translation," *CoRR*, vol. abs/1803.05567, 2018.

[3] L. Specia, K. Shah, J. G. Souza, and T. Cohn, "Quest-a translation quality estimation framework," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 79–84, 2013.

[4] A. F. T. Martins, F. Kepler, and J. Monteiro, "Unbabel's participation in the wmt17 translation quality estimation shared task," in *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, (Copenhagen, Denmark), pp. 569–574, 2017.

[5] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, *et al.*, "Findings of the 2017 conference on machine translation (wmt17)," in *Proceedings of the Second Conference on Machine Translation*, pp. 169–214, 2017.

[6] A. F. Martins, M. Junczys-Dowmunt, F. N. Kepler, R. Astudillo, C. Hokamp, and R. Grundkiewicz, "Pushing the limits of translation quality estimation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 205–218, 2017.

[7] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of association for machine translation in the Americas*, vol. 200, 2006.

[8] M. Junczys-Dowmunt and R. Grundkiewicz, "Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing," *arXiv preprint arXiv:1605.04800*, 2016.

[9] H. Kim, J.-H. Lee, and S.-H. Na, "Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation," in *Proceedings of the Second Conference on Machine Translation*, pp. 562–568, 2017.

[10] 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 Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 17, no. 1, p. 3, 2017.

[11] H. Kim and J.-H. Lee, "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*, pp. 494–498, 2016.

[12] Y. Zhang and D. Weiss, "Stack-propagation: Improved representation learning for syntax," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1557–1566, Association for Computational Linguistics, 2016.

[13] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[14] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[16] M. Rikters, M. Fishel, and O. Bojar, "Visualizing Neural Machine Translation Attention and Confidence," vol. 109, (Lisbon, Portugal), pp. 1–12, 2017.

[17] M. Rikters and M. Fishel, "Confidence through attention," in *Proceedings of MT Summit XVI*, (Nagoya, Japan), pp. 299–311, 2017.

[18] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A. V. Miceli Barone, J. Mokry, and M. Nadejde, "Nematus: a toolkit for neural machine translation," in *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, (Valencia, Spain), pp. 65–68, 2017.

[19] D. P. Kingma and L. Ba, "J. adam: a method for stochastic optimization," in *International Conference on Learning Representations*, 2015.