Unbabel's Participation in the WMT17 Translation Quality Estimation Shared Task

André F. T. Martins
Unbabel & Instituto de Telecomunicações
Lisbon, Portugal
andre.martins@unbabel.com

Fabio N. Kepler
Unbabel
University of Pampa, Alegrete, Brazil
kepler@unbabel.com

José L. Monteiro
Unbabel
Lisbon, Portugal
jose@unbabel.com

Abstract

This paper presents the contribution of the Unbabel team to the WMT 2017 Shared Task on Translation Quality Estimation. We participated on the word-level and sentence-level tracks. We describe our two submitted systems: (i) STACKEDQE, a "pure" QE system, trained only on the provided training sets, which is a stacked combination of a feature-rich sequential linear model with a neural network, and (ii) FULLSTACKEDQE, which also stacks the predictions of an automatic post-editing system, trained on additional data. When evaluated on the English-German and German-English datasets, FULLSTACKEDQE achieved word-level F_1^{MULT} scores of 56.6% and 52.9%, and sentence-level correlation Pearson scores of 64.1% and 62.6%, respectively. Our system ranked second in both tracks, being statistically indistinguishable from the best system in the word-level track.

1 Introduction

Quality estimation is the task of evaluating a translation system's quality without access to reference translations (Blatz et al., 2004; Specia et al., 2013). This paper describes the contribution of the Unbabel team to the Shared Task on Sentence-Level and Word-Level Quality Estimation (QE Tasks 1 and 2) at the 2017 Conference on Statistical Machine Translation (WMT 2017).

In the word-level task, the goal is to predict the word-level quality of machine translated text, by assigning a label of OK or BAD to each word in the translation. The sentence-level task attempts to predict the HTER of each sentence, along with a ranking of the sentences. Two language pairs and domains are considered: English-German (IT domain) and German-English (medical domain).

Our submission is largely based on the approach that we have recently proposed in Martins et al. (2017), which ensembles a "pure" quality estimation system with predictions derived from an automatic post-editing system. The focus was on developing a word-level system, and to use the word label predictions to predict the sentence-level HTER.

Our system architecture is described in full detail in the following sections. We first describe our "pure" QE system (§2), which consists of a neural model (NEURALQE) stacked into a linear feature-rich classifier (LINEARQE). Then, we train an APE system (using a large amount of artificial "roundtrip translations") and adapt it to predict word-level quality labels (yielding APEQE, §3). We show that the pure and the APE-based QE system are highly complementary (§4): our best system is a stacked combination of LINEARQE, NEURALQE, and APEQE. By employing a simple word-to-sentence conversion, we adapt our systems to sentence-level QE. Overall, we achieve word-level F_1^{MULT} scores of 56.6% and 52.9% and sentence-level Pearson scores of 64.1% and 62.6% for English-German and German-English, respectively.

The following external resources were used: part-of-speech tags and extra syntactic dependency information were obtained with TurboTagger and TurboParser (Martins et al., 2013),[1]

[1]Publicly available on http://www.cs.cmu.edu/~ark/TurboParser/.

trained on the Penn Treebank (for English) and on the version of the German TIGER corpus used in the SPMRL shared task (Seddah et al., 2014). For the neural models, we used pre-trained word embeddings from Polyglot (Al-Rfou et al., 2013).

For our FULLSTACKEDQE submission, we also use additional data to train the APE-based QE systems: for English-German, the set of 500K artificial roundtrip translations provided by Junczys-Dowmunt and Grundkiewicz (2016), and, for German-English, the UFAL Medical Corpus provided in the WMT17 Biomedical Translation task.

2 Pure Quality Estimation

We use the pure quality estimation system developed by the Unbabel team and described in Martins et al. (2017), which consists of an ensemble of a linear feature-based classifier with a neural network. We briefly describe the linear (§2.1) and neural (§2.2) components of our system, as well as their combination (§2.3). Further details are presented in Martins et al. (2016, 2017).

2.1 Linear Sequential Model

The linear component of our model is a discriminative feature-based sequential model (called LINEARQE). The system receives as input a tuple $\langle s, t, \mathcal{A} \rangle$, where $s = s_1 \dots s_M$ is the source sentence, $t = t_1 \dots t_N$ is the translated sentence, and $\mathcal{A} \subseteq \{(m, n) \mid 1 \leq m \leq M,\ 1 \leq n \leq N\}$ is a set of word alignments. It predicts as output a sequence $\widehat{y} = y_1 \dots y_N$, with each $y_i \in \{\text{BAD}, \text{OK}\}$. This is done as follows:

$$\widehat{y} = \text{argmax}_y \sum_{i=1}^{N} \boldsymbol{w}^\top \boldsymbol{\phi}_u(s, t, \mathcal{A}, y_i) + \sum_{i=1}^{N+1} \boldsymbol{w}^\top \boldsymbol{\phi}_b(s, t, \mathcal{A}, y_i, y_{i-1}). \quad (1)$$

Above, $\boldsymbol{w}$ is a vector of weights, $\boldsymbol{\phi}_u(s, t, \mathcal{A}, y_i)$ are unigram features (depending only on a single output label), $\boldsymbol{\phi}_b(s, t, \mathcal{A}, y_i, y_{i-1})$ are bigram features (depending on consecutive output labels), and y_0 and y_{N+1} are special start/stop symbols.

Table 1 shows the unigram and bigram features used in the LINEARQE system. We include features that depend on the target word and its aligned source word, as well as the context surrounding them.[2] We include also syntactic features to detect grammatically incorrect constructions. We use features that involve the dependency relation, the head word, and second-order sibling and grandparent structures. Features involving part-of-speech (POS) tags and syntactic information are obtained with TurboTagger and TurboParser (Martins et al., 2013). The feature weights are learned by running 50 epochs of the max-loss MIRA algorithm (Crammer et al., 2006), with regularization constant $C \in \{10^{-k}\}_{k=1}^{4}$ and a Hamming cost function placing a higher penalty on false positives than on false negatives ($c_{\text{FP}} \in \{0.5, 0.55, \dots, 0.95\}, c_{\text{FN}} = 1 - c_{\text{FP}}$), to account for the existence of fewer BAD labels than OK labels in the data. These values are tuned on the development set.

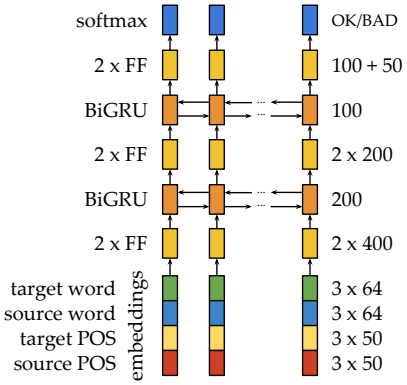

Figure 1: Architecture of our NEURALQE system.

2.2 Neural System

Next, we describe the neural component of our pure QE system, which we call NEURALQE.

The architecture of NEURALQE is depicted in Figure 1. We used Keras (Chollet, 2015) to implement our model. The system receives as input the source and target sentences s and t, their word-level alignments $\mathcal{A}$, and their corresponding POS tags obtained from TurboTagger. The input layer follows a similar architecture as QUETCH (Kreutzer et al., 2015), with the addition of POS features. A vector representing each target word is obtained by concatenating the embedding of that word with those of the aligned word in the source.[3] The immediate left and right contexts for source and target words are also concatenated. We use

[2] Features involving the aligned source word are replaced by NIL if the target word is unaligned. If there are multiple aligned source words, they are concatenated into a single feature.

[3] For the cases in which there are multiple source words aligned to the same target word, the embeddings are averaged.

Features	Label	Input (referenced by the ith target word)
unigram	$y_i \wedge \ldots$	*BIAS *WORD, LEFTWORD, RIGHTWORD *SOURCEWORD, SOURCELEFTWORD, SOURCERIGHTWORD *LARGESTNGRAMLEFT/RIGHT, SOURCELARGESTNGRAMLEFT/RIGHT *POSTAG, SOURCEPOSTAG †WORD+LEFTWORD, WORD+RIGHTWORD †WORD+SOURCEWORD, POSTAG+SOURCEPOSTAG
simple bigram	$y_i \wedge y_{i-1} \wedge \ldots$	*BIAS
rich bigrams	$y_i \wedge y_{i-1} \wedge \ldots$ $y_{i+1} \wedge y_i \wedge \ldots$	all above WORD+SOURCEWORD, POSTAG+SOURCEPOSTAG
syntactic	$y_i \wedge \ldots$	DEPREL, WORD+DEPREL HEADWORD/POSTAG+WORD/POSTAG LEFTSIBWORD/POSTAG+WORD/POSTAG RIGHTSIBWORD/POSTAG+WORD/POSTAG GRANDWORD/POSTAG+HEADWORD/POSTAG+WORD/POSTAG

Table 1: Features used in the LINEARQE system (see Martins et al., 2016 for a detailed description). Features marked with * are included in the WMT16 baseline system. Those marked with † were proposed by Kreutzer et al. (2015).

the pre-trained 64-dimensional Polyglot word embeddings (Al-Rfou et al., 2013) for English and German, and refine them during training. In addition to this, POS tags for each source and target word are also embedded and concatenated. POS embeddings have size 50 and are initialized as described by Glorot and Bengio (2010). A dropout probability of 0.5 is applied to the resulting vector representations.

The following layers are then applied in sequence:

1. Two feed-forward layers of size 400 with rectified linear units (ReLU; Nair and Hinton (2010));
2. A layer with bidirectional gated recurrent units (BiGRU, Cho et al. (2014)) of size 200, where forward and backward vectors are concatenated, trained with layer normalization (Ba et al., 2016);
3. Two feed-forward ReLU layers of size 200;
4. A BiGRU layer of size 100 with identical configuration to the previous BiGRU;
5. Two more feed-forward ReLU layers of sizes 100 and 50, respectively.

As the output layer, a softmax transformation over the OK/BAD labels is applied. We provide ablation experiments in Martins et al. (2017) to validate this architecture choice.

We train the model with the RMSProp algorithm (Tieleman and Hinton, 2012) by minimizing the cross-entropy with a linear penalty for BAD word predictions, as in Kreutzer et al. (2015). We set the BAD weight factor to 3.0. All hyperparameters are adjusted based on the development set. Target sentences are bucketed by length and then processed in batches (without any padding or truncation).

Finally, we also trained 5 independent instances of NEURALQE with different random initializations and different data shuffles. We ensemble these systems by taking the averaged probability of each word being BAD. Tables 2–3 show the results. We observe that, for both language pairs, the neural model outperforms the linear model, and the ensemble of 5 neural systems achieves an extra boost (most noticeable in the English-German dataset).

2.3 Stacking Neural and Linear Models

We now stack the NEURALQE system (§2.2) into the LINEARQE system (§2.1) as an ensemble strategy; we call the resulting system STACKEDQE.

The individual instances of the neural systems are incorporated in the stacking architecture as different features, yielding STACKEDQE. In total, we have 5 predictions (probability values given by each NEURALQE system) for every word in the training, development and test datasets. These predictions are plugged as additional features in the

LINEARQE model. As unigram features, we used one real-valued feature for every model prediction at each position, conjoined with the label. As bigram features, we used two real-valued features for every model prediction at the two positions, conjoined with the label pair.

For the remainder of this paper, we will take STACKEDQE as our pure QE system.

3 APE-Based Quality Estimation

To develop our APE-based QE system (APEQE), we followed a similar approach as the one described in Martins et al. (2017), with a few minor differences, explained below.

Junczys-Dowmunt and Grundkiewicz (2016) applied neural translation models to the APE problem, treating different models as components in a log-linear model, allowing for multiple inputs (the source s and the translated sentence t) that were decoded to the same target language (post-edited translation p). Two systems were considered, one using s as the input ($s \rightarrow p$) and another using t as the input ($t \rightarrow p$). For English-German, we used the 500K artificial roundtrip translations provided by the shared task organizers, along with the original data from the shared task (oversampled 20 times, as in Junczys-Dowmunt and Grundkiewicz (2016)). For German-English, we only considered the $s \rightarrow p$ machine translation system, trained from a subset of the UFAL Medical Corpus provided in the WMT17 Biomedical Translation task. This subset was obtained through cross-entropy filtering. For this, we built an in-domain trigram language model from the English post-edited training data. We then calculated cross-entropy scores for the UFAL corpus according to the language model. We sorted the corpus by increasing cross-entropy and kept the first 500K sentences to be used as additional training data.

To convert the resulting APE systems into word-level quality estimators, we need to turn the automatic post-edited sentences into word quality labels. This is done in a straightforward way by using the TERCOM software tool (Snover et al., 2006)[4] with the default settings (tokenized, case insensitive, exact matching only, shifts disabled). This tool computes the HTER (the normalized edit distance) between the translated and post-edited sentence. As a by-product, it aligns the words in the two sentences, identifying substitution errors, word deletions (i.e. words omitted by the translation system), and insertions (redundant words in the translation). This is mapped deterministically into OK and BAD labels.

Our approach for the shared task differs from Martins et al. (2017) in which we skipped the QE-tuning step when training the log-linear APE model; instead, we kept the output of the $s \rightarrow p$ and the $t \rightarrow p$ systems, converted each to word-level quality labels, and then include the two predictions as additional features in the FULLSTACKEDQE system, described below. We denote the individual systems as APEQE $s \rightarrow p$ and APEQE $t \rightarrow p$, and the combined system as FULLSTACKEDQE $s, t \rightarrow p$.

4 Full Stacked System

Finally, we consider a larger stacked system where we stack both NEURALQE and APEQE into LINEARQE. This mixes pure QE with APE-based QE systems; we call the result FULLSTACKEDQE. The procedure is analogous to that described in §2.3, with extra binary features for the APE-based word quality label predictions. For training, we used jackknifing to ensure the predictions on the training set are not biased.

4.1 Word-Level QE

The performance of the FULLSTACKEDQE system on the English-German and German-English development datasets are shown in Tables 2–3. For English-German, we compare with the system from Martins et al. (2017).

For both language pairs, we can see that the APE-based and the pure QE systems are highly complementary. For English-German, the full combination of the linear, neural, and APE-based systems improves the scores with respect to the best individual system by about 5.5 points. There is a small improvement by including also a feature from APE $t \rightarrow p$, in addition to $s \rightarrow p$. For German-English, we observe an improvement of 4.9 points.

4.2 Sentence-Level QE

We followed the same procedure of Martins et al. (2017) to convert word-level quality predictions to a sentence-level HTER prediction. For the APE system, we simply measured the HTER between the translated sentence t and the predicted corrected sentence $\widehat{p}$. For a pure QE system, we ap-

[4] http://www.cs.umd.edu/~snover/tercom.

	F_1^{MULT} dev	F_1^{MULT} test 2016
Martins et al. (2017)	56.80	57.47
LINEARQE	47.71	48.09
NEURALQE (single)	49.58	49.95
NEURALQE (5-avg)	51.37	51.38
STACKEDQE	52.72	52.89
APEQE ($s \rightarrow p$)	51.43	52.47
APEQE ($t \rightarrow p$)	35.27	37.13
FULLSTACKEDQE ($s \rightarrow p$)	57.18	58.04
FULLSTACKEDQE ($s, t \rightarrow p$)	**57.55**	**58.36**

Table 2: Performance of the several word-level QE systems on the development and WMT16 English-German test datasets.

	F_1^{MULT} dev
LINEARQE	48.07
NEURALQE (single)	49.39
NEURALQE (avg-5)	49.58
STACKEDQE	53.22
APEQE ($s \rightarrow p$)	45.09
FULLSTACKEDQE ($s \rightarrow p$)	**58.08**

Table 3: Performance of the several word-level QE systems on the German-English development dataset.

plied the following word-to-sentence conversion technique: (i) run a QE system to obtain a sequence of OK and BAD word quality labels; (ii) use the fraction of BAD labels as an estimate for HTER. Finally, to combine the APE and pure QE systems toward sentence-level QE, we simply take the average of the two HTER predictions above.

Table 4 shows the results obtained with our pure QE system (STACKEDQE), with our APE-based system (APEQE), and with the combination of the two (FULLSTACKEDQE). We report also the performance of the system of Martins et al. (2017) for English-German, for comparison.

5 Final Results

Finally, we show in Tables 5–6 the results obtained in the test set for our two submitted systems, STACKEDQE and FULLSTACKEDQE, in word-level and sentence-level quality estimation. As expected, the inclusion of the predictions made by the APE system gave a significant boost for the word-level task (>5 F_1^{MULT} points for English-German, and >6 points for German-English) and for the sentence-level task (>5 Pearson correlation points for English-German, >4 points for German-English).

6 Conclusions

We have presented the contribution of the Unbabel team to the WMT 2017 Shared Task on Translation Quality Estimation. Our word-level system combines a pure quality estimation system, based on stacking a neural and feature-based linear model, and an APE-based quality estimation system, which uses the predictions of an automatic post-editing system to generate additional features. We applied a simple conversion strategy to obtain a sentence-level quality estimator based on the word-level one. The system is evaluated on two language pairs, English-German and German-English.

Acknowledgments

This work was partially supported by the the EXPERT project (EU Marie Curie ITN No. 317471), and by Fundação para a Ciência e Tecnologia (FCT), through contracts UID/EEA/50008/2013 and UID/CEC/50021/2013, the LearnBig project (PTDC/EEI-SII/7092/2014), and the GoLocal project (grant CMUPERI/TIC/0046/2014).

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662* .

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* .

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proc. of the International Conference on Computational Linguistics*. page 315.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In *Proc. of Empirical Methods in Natural Language Processing*.

François Chollet. 2015. Keras. https://github.com/fchollet/keras.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research* 7:551–585.

		Pearson dev	Pearson test 2016	Spearman dev	Spearman test 2016
En-De	Martins et al. (2017)	64.04	**65.56**	65.52	65.92
	STACKEDQE	62.18	59.90	64.07	60.23
	APEQE	58.79	56.69	59.41	59.43
	FULLSTACKEDQE	**64.33**	65.25	**65.65**	**66.10**
De-En	STACKEDQE	60.47	–	59.74	–
	APEQE	59.94	–	57.68	–
	FULLSTACKEDQE	**67.97**	–	**62.45**	–

Table 4: Performance of our sentence-level QE systems on the English-German and German-English datasets, as measured by the official evaluation script. We show the performance of Martins et al. (2017) for comparison.

	F_1^{OK}	F_1^{BAD}	F_1^{MULT}
EN-DE STACKEDQE	88.2	58.1	51.2
EN-DE FULLSTACKEDQE	90.6	62.5	56.6
DE-EN STACKEDQE	93.6	49.7	46.6
DE-EN FULLSTACKEDQE	94.1	56.2	52.9

Table 5: Performance of the submitted word-level systems on the test set.

	Pearson	Spearman
EN-DE STACKEDQE	58.9	61.0
EN-DE FULLSTACKEDQE	64.1	65.2
DE-EN STACKEDQE	58.0	57
DE-EN FULLSTACKEDQE	62.6	61

Table 6: Performance of the submitted sentence-level systems on the test set.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*. pages 249–256.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation*. pages 751–758.

Julia Kreutzer, Shigehiko Schamoni, and Stefan Riezler. 2015. Quality estimation from scratch (quetch): Deep learning for word-level translation quality estimation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. pages 316–322.

André F. T Martins, Miguel B. Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

André F. T. Martins, Ramón Astudillo, Chris Hokamp, and Fábio Kepler. 2016. Unbabel's participation in the wmt16 word-level translation quality estimation shared task. In *Proceedings of the First Conference on Machine Translation*.

André F. T. Martins, Marcin Junczys-Dowmunt, Fabio Kepler, Ramon Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics (to appear)* .

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proc. of the International Conference on Machine Learning*. pages 807–814.

Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. 2014. Introducing the SPMRL 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*. pages 103–109.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of the 7th Conference of the Association for Machine Translation in the Americas*. pages 223–231.

Lucia Specia, Kashif Shah, Jose G.C. de Souza, and Trevor Cohn. 2013. QuEst - a translation quality estimation framework. In *Proc. of the Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. pages 79–84. http://www.aclweb.org/anthology/P13-4014.

Tijmen Tieleman and Geoffrey Hinton. 2012. Rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning* 4(2).