Bilexical Embeddings for Quality Estimation

Frédéric Blain, Carolina Scarton and **Lucia Specia**
Department of Computer Science
University of Sheffield, UK
{f.blain, c.scarton, l.specia}@sheffield.ac.uk

Abstract

This paper describes the SHEF submissions for the three sub-tasks of the Quality Estimation shared task of WMT17, namely: (i) a word-level prediction system using bilexical embeddings, (ii) a phrase-level labelling approach based on the word-level predictions, (iii) a sentence-level prediction system using word embeddings and handcrafted baseline features. Results are promising for the sentence-level approach, but still very preliminary for the other two levels.

1 Introduction

Quality Estimation (QE) allows the evaluation of Machine Translation (MT) when reference translations are not available. It can be used in various ways such as in post-editing (PE) to predict whether or not an automatically generated sentence is worth publishing, editing or it should be retranslated manually. Word-level predictions can be helpful by highlighting words that cannot be relied upon or should be fixed by post-editors. More recently, QE at phrase-level has emerged as a way of using quality predictions at decoding time in phrase-based Statistical MT (SMT) systems to guide the decoder such as to keep phrases which are predicted as good, and conversely to discard those which are predicted as bad (Logacheva, 2017).

QE models are built based on a list of features along with a Machine Learning algorithm for either regression or classification. These features are usually extracted from the source and target texts or from the MT system that generated the translations. Shah et al. (2015) introduced a new set of features extracted using an unsupervised approach with the use of neural network: continuous-space language model features and word embeddings features.

In our contribution this year we investigate whether we can go beyond engineered features by learning bilexical operators over distributional representations of words in source-target text pairs. Considering the MT pipeline as a noisy black-box, our motivation is to be able to build QE models to predict if information encoded in the source sentence is preserved in the target sentence after translation.

2 Bilinear Model

Madhyastha et al. (2014) propose to use word-level embeddings to predict the strength of different types of lexical relationships between a pair of words, such as head-modifier relations between noun-adjective pairs. They designed a supervised framework for learning bilexical operators over distributional representations, based on learning bilinear forms W. We adapted their method to predict the strength of relationship between source and target words. This problem is formulated as a log-bilinear model, parametrized with W as follows:

$$\Pr(t|s;W) = \frac{\exp\left\{\phi(t)^\top W \phi(s)\right\}}{\sum_{t' \in \mathcal{T}} \exp\left\{\phi(t')^\top W \phi(s)\right\}} \quad (1)$$

where ϕ denotes the word embeddings of any given word in a vocabulary $\mathcal{V}$. The source words s and target words t are respectively taken from subspaces $\mathcal{S} \subseteq \mathcal{V}$ and $\mathcal{T} \subseteq \mathcal{V}$.

In essence, the problem can be reduced to first obtaining the corresponding word embeddings of the vocabularies of both source and target sentences using a substantially large monolingual corpus for each of the two languages, followed by using the bilinear model to estimate W. W is learned

	IT		PHARMA	
	#sent	#word	#sent	#word
English	3.4M	58.3M	1.8M	78.5M
German	3.4M	57.5M	1.8M	83.6M

Table 1: Statistics of the in-domain data used to train our embeddings.

using the source-target word *alignment* by minimizing the negative log-likelihood using a ℓ_2 regularized objective as:

$$L(W) = -\sum_{s,t} \log(Pr(t|s;W)) + \lambda \|W\|_2 \quad (2)$$

where λ is the constant that controls the capacity of W with gradient descent-based optimization.

We explore this approach for both word and phrase-level QE. For training, we rely on both the word-alignments and the gold QE labels (i.e. the OK/BAD labels). The former gives us the source-target pairs, and the latter whether this pair is valid or not. Our assumption is that this approach should be able to predict whether or not a word in the target language (MT output) is correct by exploring the strength of the linguistic relation with the source word it is generated from.

3 Experimental Settings

3.1 Data and Gold labels

Each QE shared task has two datasets: English→German segments on the IT domain (with 23,000 sentences for training, 1,000 for development and 2,000 for test), and German→English segments on the Pharmaceutical domain (with 25,000 sentences for training, 1,000 for development and 2,000 for test). The same data is used for all three tasks: word, phrase and sentence-level prediction.

For the word-level task, each token of the MT is annotated with OK or BAD labels. For the phrase-level task, phrases are segmented as given by an SMT decoder and also annotated with OK or BAD labels. Finally, for the sentence-level task, the quality label is a Human-Targeted Error Rate (HTER) score (Snover et al., 2009).

3.2 Word Embeddings

Word embeddings were used in our submissions for the three tasks. We trained in-domain skip-gram embeddings on the in-domain data shown in Table 1 using FastText[1] (Bojanowski et al., 2016) with 300 dimensions and learning rate set to 0.025. The default training settings are otherwise used. The in-domain data is the same as that used to train the SMT system that produced the translations in the QE datasets, as made available by the task organizers.

For the word and phrase-level tasks, we used our word embeddings to obtain a word vector representation of 300 dimensions for each word of both the training and development sets. For the sentence-level task, the word embeddings are averaged for each sentence, as previously applied in (Scarton et al., 2016).

3.3 Tool

To learn to predict the labels for the word-level task, we used BMAPS[2], the toolkit implementing the method in (Madhyastha et al., 2014) along with the word alignments provided by the organizers (as produced by the SMT system). BMAPS is used to learn the bilexical operators between both source and target embeddings. The tool relies on three matrices corresponding to the source and target vocabularies of the training data, and a third matrix representing the word-level lexical relation between them. This matrix is built from the word-level alignments and the gold labels to indicate which lexical items form a pair, and whether their lexical relation is OK or BAD (i.e. if two lexical items are aligned and labelled as OK, their intersection in the third matrix is set to 1, 0 otherwise).

By default, the model is trained over 100 iterations with the l_2 norm as regularizer, and using the *forward-backward splitting* algorithm (FOBOS) (Duchi and Singer, 2009) as optimization scheme ($lc = 0.1$, $tau = 0.1$).

3.4 Evaluation

We used the official task metrics to evaluate our results. For the word and phrase-level tasks, the metrics are F_1-BAD and F_1-OK which correspond to the F_1 scores on both BAD and OK labels, and F_1-multi which is the product of the two formers. For the sentence-level task, the metrics for scoring are Pearson's correlation (primary metric), Mean Average Error (MAE) and Root Mean Squared Error (RMSE), and for ranking, Spearman's rank correlation (primary metric) and DeltaAvg.

[1] https://github.com/facebookresearch/fastText

[2] https://github.com/f00barin/bmaps

4 Results

4.1 Word-level QE prediction (Task 2)

We investigate different context windows to build our lexical representations, ranging from a wide window considering all sentence-level context, to a much narrower approach representing each word individually:

- **Full context:** each word is associated with its left and right context to capture the exact distributional features of the specific context in which this lexical item occurs. A lexical item is thus a 900-dimensional word vector represented by the tuple $< emb_{left}, emb_{cur}, emb_{right} >$, where emb_{left} and emb_{right} are the averaged embeddings of the left/right contexts and emb_{cur} the word representation of the current word. Here our assumption is that a lexical item would represent a word within its context and at its position in the sentence, therefore if the word appears twice in the sentence, it would be represented by two different lexical items.

- **Surrounding context:** instead of considering all the left and right context of the current word, we limit ourselves to the two surrounding words. This allows for a model that is as generic as possible while still considering two distributional features corresponding to two different lexical items. Here the assumption is the same as before, the lexical item which represents a word is the same but only considering a window of one word on the left/right to compute emb_{left}/emb_{right}.

- **Unigram:** we use only the embeddings of the current word without considering any surrounding context. By doing so, we fully rely on the embeddings and the way they are trained (skipgram). In this case, the lexical item is a single word representation of 300 dimensions.

For each context we investigate two variants: with and without the use of the gold labels in order to demonstrate the capacity of our approach to learn how to discriminate the valid lexical pairs from the others.

Discussion The results of our approach for the word-level task are given in Table 2. We report the results of our official submissions to the task (†) along with additional experiments we conducted after the task deadline. They are both compared with the official baseline of Task 2.

Our first observation is the overall low performance of our approach compared to the official baseline. However, we found very encouraging the results of our additional experiments compared to those of the systems submitted. The revised training procedure significantly improved the performance in terms of F_1-OK for all three contexts types, resulting in a boost in the F_1-multi scores.

To better understand the gap between our official and additional results, it is important to mention the technical constraints we faced performing the task with BMAPS for the official submission. In its current implementation, BMAPS relies on non-sparse matrices which in our case lead to a heavy memory print, since the source and the target matrices contain vector representations for each word in the corpus. Therefore, to be able to run BMAPS on our servers we were limited to use up to 2,000 sentences (about 9% of the training corpus) as training instances. This certainly had a significant impact on the performance of the models.

To tackle this constraint we later opted for a mini-batch training approach: we divided the training corpus into batches of 500 sentences, the training for each batch starting from the results from the training with the previous one. By doing so we are able to use all the training data. However, in BMAPS the size of the dev set (in terms of words from which the matrices are built) has to be smaller than that of the training set. Therefore, by using mini-batches we had to reduce our dev set. We selected for the dev set 250 sentences with the highest number of `OK` labels in order to boost performance for this class. We also refined our training parameters by switching to the nuclear norm (which is expected to converge faster when restricting the training size (Madhyastha et al., 2014)). Finally, we empirically identified the best values for the two main parameters (namely *lc* and *tau*) for different context types: for both the full and surrounding context, we used $lc = 0.1$ and $tau = 0.001$, while for the unigram approach we used $lc = 0.1$ and $tau = 0.01$.

As a second finding, one can notice the impact of considering the surrounding context when predicting each word's label. In both official and additional results, there is a substantial difference be-

	norm	training size	F_1-BAD	F_1-OK	F_1-**multi**
English→German (2016)					
BMAPS-full	l_2	2k	0.326	0.103	0.034
BMAPS-nolabel-full	l_2	2k	0.311	0.222	0.069
BMAPS-full	$nuclear$	23k	0.321	0.817	0.262
BMAPS-window	l_2	2k	0.328	0.207	0.068
BMAPS-nolabel-window	l_2	2k	0.315	0.170	0.053
BMAPS-window	$nuclear$	23k	0.325	0.819	0.266
BMAPS-unigram †	l_2	2k	0.316	0.501	0.158
BMAPS-nolabel-unigram †	l_2	2k	0.296	0.330	0.098
BMAPS-unigram	$nuclear$	23k	0.251	0.845	0.212
BASELINE	–	–	0.404	0.892	0.360
English→German (2017)					
BMAPS-full	$nuclear$	23k	0.336	0.812	0.273
BMAPS-window	$nuclear$	23k	0.343	0.812	0.279
BMAPS-unigram †	l_2	2k	0.325	0.484	0.157
BMAPS-nolabel-unigram †	l_2	2k	0.302	0.322	0.097
BMAPS-unigram	$nuclear$	23k	0.270	0.848	0.229
BASELINE	–	–	0.407	0.886	0.361
German→English (2017)					
BMAPS-full	$nuclear$	25k	0.231	0.447	0.103
BMAPS-window	$nuclear$	25k	0.235	0.506	0.119
BMAPS-unigram †	l_2	2k	0.210	0.419	0.088
BMAPS-nolabel-unigram †	l_2	2k	0.209	0.391	0.082
BMAPS-unigram	$nuclear$	25k	0.234	0.527	0.123
BASELINE	–	–	0.365	0.939	0.342

Table 2: Results of our word-level predictions. † denotes our official submissions to the task using the l_2 norm and single training set of 2k sentences. The other figures are obtained with mini-batch training using 500 sentences at the time. In grey are the results of the official baseline of the task.

tween the three types of context: while unigram was the best performing when limited to 2k training instances only, the exact opposite was found when using the full training set with better F_1-* scores when the context in which the word occurs is employed. Furthermore, we note a small advantage for the window context over the full context in both language pairs. We believe this means that considering the surrounding context could better help in a situation where a word would appear twice in the same sentence but should be labelled differently.

Overall, these results are encouraging and we aim to pursue further investigations towards improving this approach for the task of word-level QE.

4.2 Phrase-level QE labelling (Task 3)

While we could have chosen to predict phrase-level QE labels similarly to our word-level predictions, we opted for generating phrase-level labels from word-level labels following the labelling approaches described in Blain et al. (2016):

- **Optimistic:** if half or more of words have a label `OK`, the phrase has the label `OK` (majority labelling).
- **Pessimistic:** if 30% words or more have a label `BAD`, the phrase has the label `BAD`.
- **Super-pessimistic:** if any word in the phrase has a label `BAD`, the whole phrase has the label `BAD`.

Discussion The results of these three phrase-level labelling strategies based upon our word-level predictions are given in Table 3. We report the results of our official submissions to the task (†) along with additional experiments we conducted after the task deadline. These are compared with the official baseline for Task 3.

First, similarly to the word-level task, the performance at phrase-level improved with the additional experiments, which was expected since the labelling directly follows from the word-level predictions. Second, while we originally observed better labelling performance using the optimistic approach on test.2016 (see underlined numbers), we now observe better F_1-* scores with both pessimistic approaches for en→de. One can also observe comparable performance for en→de when the surrounding context is used: the difference in terms of F_1-* scores between the full and window context is marginal. For de→en this is different: the phrase labelling based on word predictions using the window context outperforms the phrase la-

	F_1-BAD	F_1-OK	F_1-**multi**
English→German (2016)			
BMAPS-full-opti	0.292	0.799	0.233
BMAPS-window-opti	0.284	0.798	0.227
BMAPS-unigram-opti †	0.415	0.562	0.233
BMAPS-unigram-nolabel-opti †	0.398	0.373	0.149
BMAPS-unigram-opti	0.166	0.816	0.135
BMAPS-full-pess	0.425	0.743	0.316
BMAPS-window-pess	0.426	0.742	0.316
BMAPS-unigram-pess •	0.452	0.264	0.120
BMAPS-unigram-nolabel-pess •	0.442	0.140	0.062
BMAPS-unigram-pess	0.341	0.780	0.266
BMAPS-full-superpess	0.441	0.723	0.318
BMAPS-window-superpess	0.437	0.719	0.314
BMAPS-unigram-super-pess •	0.455	0.250	0.114
BMAPS-unigram-nolabel-suppess •	0.442	0.136	0.060
BMAPS-unigram-super-pess	0.366	0.763	0.279
BASELINE	0.403	0.812	0.328
English→German (2017)			
BMAPS-full-opti	0.309	0.804	0.248
BMAPS-window-opti	0.312	0.800	0.250
BMAPS-unigram-opti †	0.409	0.553	0.226
BMAPS-unigram-nolabel-opti †	0.388	0.380	0.148
BMAPS-unigram-opti	0.184	0.823	0.152
BMAPS-full-pess	0.431	0.750	0.323
BMAPS-window-pess	0.428	0.743	0.318
BMAPS-unigram-pess	0.350	0.794	0.278
BMAPS-full-super-pess	0.438	0.733	0.321
BMAPS-window-super-pess	0.437	0.724	0.316
BMAPS-unigram-super-pess	0.368	0.781	0.287
BASELINE	0.402	0.814	0.327
German→English (2017)			
BMAPS-full-opti	0.326	0.478	0.156
BMAPS-window-opti	0.334	0.565	0.189
BMAPS-unigram-opti †	0.299	0.473	0.141
BMAPS-unigram-nolabel-opti †	0.300	0.440	0.132
BMAPS-unigram-opti	0.336	0.593	0.199
BMAPS-full-pess	0.313	0.281	0.088
BMAPS-window-pess	0.320	0.357	0.114
BMAPS-unigram-pess	0.322	0.378	0.122
BMAPS-full-super-pess	0.311	0.256	0.079
BMAPS-window-super-pess	0.317	0.332	0.106
BMAPS-unigram-super-pess	0.320	0.358	0.115
BASELINE	0.397	0.907	0.360

Table 3: Results of the phrase-level labelling strategies based upon our word-level QE predictions. † denotes our official submissions to the task and • the results of the other two labelling strategies, both using our official submissions to Task 2. The other figures are obtained with the updated word predictions from Task 2 resulting of the full batch training. In grey are the results of the official baseline of the task.

belling based on word prediction using the entire sentence as context.

4.3 Sentence-level QE prediction (Task 1)

For the sentence-level task we followed a simple approach, which had been previously applied by Scarton et al. (2016) for document-level QE. The idea was to combine word embeddings with handcrafted features.

However, whilst Scarton et al. (2016) have used word embeddings trained on general purpose data, our embeddings are trained over in-domain data, as previously described. Word embeddings were averaged at sentence level in order to have a single vector representing each sentence. We then concatenated source and target in-domain embeddings with the 17 sentence-level baseline features provided by the organisers. An SVM regressor was used to train our QE model with hyper-parameters optimized via grid-search. For that we used the learning module available at `QuEst++` toolkit (Specia et al., 2015).

Although the sentence-level experiment is different from the approach applied for word and phrase-level tasks, our aim was to test the usability of the in-domain word embeddings. Our results are compared with the official baseline.

	Scoring		Ranking	
	Pearson's r	MAE	Spearman's ρ	DeltaAvg
English→German (2016)				
QUEST-EMB	0.50	0.12	0.53	9.02
BASELINE	0.40	0.13	0.44	7.42
English→German (2017)				
QUEST-EMB	0.50	0.13	0.51	8.96
BASELINE	0.40	0.14	0.43	7.45
German→English (2017)				
QUEST-EMB	0.56	0.12	0.56	8.79
BASELINE	0.44	0.13	0.45	6.81

Table 4: Results of QUEST-EMB in the sentence-level QE task. In grey are the results of the official baseline of the task.

Discussion The results of our sentence-level predictions are given in Table 4. Although the approach is rather simplistic, it achieves considerably good results by outperforming the baseline system and several other systems that participated in the shared task. For German→English, our system performed seventh out of 13 in the scoring task. For English→German, it performed eighth out of 13. Table 4 shows the results of our systems (called QUEST-EMB) for the different language pairs and for both scoring and ranking tasks. We also show the results of the baseline systems for comparison.

5 Conclusions

In this paper we report our submissions to the three sub-tasks of the QE campaign of WMT17. We obtained reasonably good results for the sentence-level task despite the use of a very simplistic approach. On the other hand, we significantly underperform in the two other tasks, which exploit

a bilinear model. Due to limitations regarding the experimental settings of the tool used for the official submissions, it is difficult to conclude whether or not our approach is suitable for the task of QE. In follow up experiments with different training strategies, the results proved substantially better and much more promising, albeit still behind the official baseline. This is particularly encouraging considering that the approach only relies on word embeddings and word alignment information. We plan to further experiment with it and identify possible improvements in BMAPS that could lead to better performance.

It is also worth emphasizing that the approach employed for the sentence-level task is not directly comparable to the approach used for the other tasks; they only share the embeddings trained using in-domain data. However, we can conclude that the in-domain embeddings encode useful information for all tasks.

Acknowledgments

The authors would like to thank Pranava S. Madhyastha for his support regarding the use of BMAPS. This work was supported by the QT21 project (H2020 No. 645452).

References

Frédéric Blain, Varvara Logacheva, and Lucia Specia. 2016. Phrase level segmentation and labelling of machine translation errors. In *Tenth International Conference on Language Resources and Evaluation*. Portoroz, Slovenia, pages 2240–2245.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* .

John Duchi and Yoram Singer. 2009. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research* 10(Dec):2899–2934.

Varvara Logacheva. 2017. *Human Feedback in Statistical Machine Translation*. Ph.D. thesis, The University of Sheffield.

Pranava Swaroop Madhyastha, Xavier Carreras, and Ariadna Quattoni. 2014. Learning task-specific bilexical embeddings. In *Proceedings the 25th International Conference on Computational Linguistics*. Dublin, Ireland, pages 161–171. http://www.aclweb.org/anthology/C14-1017.

Carolina Scarton, Daniel Beck, Kashif Shah, Karin Sim Smith, and Lucia Specia. 2016. Word embeddings and discourse information for Quality Estimation. In *Proceedings of the First Conference on Machine Translation*. Berlin, Germany, pages 831–837.

Kashif Shah, Varvara Logacheva, Gustavo Paetzold, Frédéric Blain, Daniel Beck, Fethi Bougares, and Lucia Specia. 2015. Shef-nn: Translation quality estimation with neural networks. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. pages 342–347.

Matthew Snover, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. 2009. Fluency, Adequacy, or HTER? Exploring Different Human Judgments with a Tunable MT Metric. In *The Fourth Workshop on Statistical Machine Translation*. Athens, Greece, pages 259–268.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*. Association for Computational Linguistics and The Asian Federation of Natural Language Processing, Beijing, China, pages 115–120. http://www.aclweb.org/anthology/P15-4020.