deepQuest: A Framework for Neural-based Quality Estimation

Julia Ive[1] **Frédéric Blain**[2] **Lucia Specia**[2]
King's College London, IoPPN, London, SE5 8AF, UK[1]
Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello, Sheffield, S1 4DP, UK[2]
julia.ive@kcl.ac.uk, {f.blain, l.specia}@sheffield.ac.uk

Abstract

Predicting Machine Translation (MT) quality can help in many practical tasks such as MT post-editing. The performance of Quality Estimation (QE) methods has drastically improved recently with the introduction of neural approaches to the problem. However, thus far neural approaches have only been designed for word and sentence-level prediction. We present a neural framework that is able to accommodate neural QE approaches at these fine-grained levels and generalize them to the level of documents. We test the framework with two sentence-level neural QE approaches: a state of the art approach that requires extensive pre-training, and a new light-weight approach that we propose, which employs basic encoders. Our approach is significantly faster and yields performance improvements for a range of document-level quality estimation tasks. To our knowledge, this is the first neural architecture for document-level QE. In addition, for the first time we apply QE models to the output of both statistical and neural MT systems for a series of European languages and highlight the new challenges resulting from the use of neural MT.

1 Introduction

Quality Estimation (QE) (Blatz et al., 2004; Specia et al., 2009) aims at predicting the quality of machine translation (MT) without human intervention. Most recent work has focused on QE to predict sentence-level post-editing (PE) effort, i.e. the process of manually correcting MT output to achieve publishable quality (Bojar et al., 2014; Bojar et al., 2015; Bojar et al., 2016a; Bojar et al., 2017). In this case, QE indicates to what extent a MT sentence needs post-editing. Document-level QE, on the other hand, scores or ranks documents according to their quality for fully automated MT usage scenarios where no post-editing can be performed, e.g. MT for gisting of news articles online.

Recently, neural methods have been successfully exploited to improve QE performance. These methods mostly rely on either complex architectures, require extensive pre-training, or need some feature engineering (Patel and M, 2016; Kim et al., 2017a; Martins et al., 2017a; Jhaveri et al., 2018). In addition, these methods have only been developed for word, phrase and sentence-level QE. These cannot be directly used for document-level QE since this level requires to take into account the content of the document in its entirety. State-of-the-art document-level QE solutions still rely on non-neural methods, and extensive feature engineering (Scarton et al., 2016).

In this paper we propose a neural framework that is able to accommodate any QE approach at a fine-grained level (e.g. a sentence-level approach), and to generalize it to learn document-level QE models. We test the framework using a state of the art neural sentence-level QE approach (Kim et al., 2017b), which uses a complex architecture and requires resource-intensive pre-training, and a light-weight neural approach employing simple encoders and no pre-training. Our sentence-level prediction approach leads to comparable or better results than the state of the art at a much lower cost. Additionally, the document-level framework improves over previous work by a large margin. To our knowledge, this is the first attempt at document-level QE using purely neural methods.

Proceedings of the 27th International Conference on Computational Linguistics, pages 3146–3157
Santa Fe, New Mexico, USA, August 20-26, 2018.

The majority of existing QE solutions for all prediction levels have been designed and tested for Statistical MT (SMT). Popular features extracted from SMT translation models are clearly no longer applicable to neural MT (NMT), while MT system-independent features, such as target language model probabilities, are likely to be less effective. For the first time, we experiment with sentence and document-level QE methods on the output of both SMT and NMT, for a series of European languages. We show that the main challenge for QE of high-quality NMT lies in detecting errors in otherwise generally fluent text. We focus on the estimation of MT quality for news texts, a type of text where gisting is seen as a popular use of MT.

We start by discussing related work in Section 2. We present our light-weight hierarchical neural QE architecture in Section 3. We then introduce our experimental settings in Section 4. We provide the results of state of the art sentence-level methods in Section 4.1 and of the proposed document-level framework in Section 4.2.

2 Related work

QE targets the prediction of MT quality in the absence of reference translations. Given a set of training examples labelled for quality, their features extracted from source units and their corresponding MT units (black-box, system-independent), optionally complemented with features related to the translation process itself (glass-box, system-dependent), a QE model can be trained to predict a score for unseen MT units. Various types of units are possible: documents, paragraphs, sentences, words and phrases have been studied in previous work to different extents. Most work has focused on sentence or word-level prediction, which have clear application in dissemination scenarios, such as MT followed by post-editing. Document-level QE, which is applicable in assimilation (i.e. gisting) scenarios, has received much less attention.

Recently, neural methods have been successfully exploited to improve QE performance. The best-performing system at the WMT 2017 shared task on QE (Bojar et al., 2017) for the three levels of prediction (word, phrase and sentence), namely `POSTECH`, is purely neural and does not rely on feature engineering (Kim et al., 2017b). `POSTECH` is a modular architecture that revolves around an encoder-decoder Recurrent Neural Network (RNN) (so-called predictor), stacked with an bidirectional RNN (so-called estimator) that produces quality estimates. It predicts quality using the weights assigned by the predictor to the words we seek to evaluate, which are concatenated with the representations of their left and right one-word contexts, and then used to feed the estimator. To perform multi-level predictions, `POSTECH` relies on a multi-task learning approach which makes the quality estimates, for different levels of prediction, interdependent. The highest level of quality labels reported by the `POSTECH` system at WMT 2017 was sentence-level. Note that, to be effective, this architecture has to be pre-trained using a significant amount of parallel data, which leads to high training requirements in terms of time, processing power and dataset size.

Jhaveri et al. (2018) propose a series of neural models for sentence-level QE by simplifying and extending the `POSTECH` architecture (e.g. they skip the Predictor step, use convolutional encoders or an additional attention mechanism). We propose an alternative RNN-based simplification of `POSTECH`.

Another well performing system in the WMT shared QE task, Unbabel (Martins et al., 2017a; Martins et al., 2017b), also uses an encoder-decoder architecture with bidirectional RNN layers as part of its stacked architecture. It follows a hybrid approach where the input to this encoder-decoder is a pre-extracted feature set: pre-trained word and part-of-speech embeddings, word alignments and contexts. The system was designed for word and sentence QE.

State-of-the-art QE solutions specifically designed for document-level prediction employ traditional machine learning algorithms with non-linear kernels, such as Support Vector Machines (Cortes and Vapnik, 1995) and Gaussian Process (Rasmussen and Williams, 2005). Standard sets of document-level features are largely inspired by sentence-level features (Bojar et al., 2016a). Additionally, various discourse and neural-based features have been explored (Scarton et al., 2016).

Document-level QE is traditionally framed as averaging over sentence-level QE (Scarton et al., 2016). Sentence-level architectures consider each sentence separately; at the document level the entirety of

sentences in the document and the importance of each of these sentences should be taken into account. While the first problem can be addressed by, for instance, merging all sentences in a document and reusing a sentence-level QE system, the second problem requires considering every sentence separately yet as a part of the document, which requires a different QE architecture.

In this work we take advantage of the ability of neural networks to capture hierarchical structures, and propose a neural framework able to generalize over any sentence-level QE approach to produce document-level QE models. We test the framework using the state of the art neural sentence-level QE approach of Kim et al. (2017b), and a low-cost neural approach employing simple encoders, which we propose.

To our knowledge, the only attempt to shed some light on QE for NMT output is that by Rikters and Fishel (2017). They use attention mechanism distributions as an indicator the confidence of the neural decoder on its output at word-level. The hypothesis is that "good" translations can be characterized by strongly focused attention connections. However, this internal information has not been proved to map directly into translation quality: a very weak correlation with human judgements in a small-scale assessment was reported. Therefore, this is the first time that experiments are performed with fully fledged, MT system-independent QE models for NMT.

3 A neural-based architecture for QE

Our framework performs multi-level translation quality prediction, which has been shown to be successful in both traditional feature-engineered QE frameworks, such as QuEst++ (Specia et al., 2015), and neural QE architectures (Kim et al., 2017a; Martins et al., 2017a). In such architectures, the representations at a given level rely on representations from more fine-grained levels (i.e. sentences for document, and words for sentence).

This is motivated by the nature of the task at hand: a document that is composed of high quality sentences is likely to have high quality as well. However, simply aggregating sentence-level predictions is not a good strategy, as a document needs to be cohesive and coherent as a whole, i.e. sentences cannot be considered completely in isolation, and thus the need of a multi-level architecture that is trained jointly arises. Another important feature of document-level prediction is that certain parts of a document may be more important than others, such sentences containing keywords in a news article, versus sentences containing background information. Therefore one should also attempt to assess whether those sentences in particular are translated accurately. We do so by using different sentence-level weighting schemes for labelling documents, and by relying on an attention mechanism.

In what follows, we will first present the two sentence-level architectures we employ to then introduce our document-level framework.

3.1 Sentence-level architectures

The encoder-decoder approach (Sutskever et al., 2014; Bahdanau et al., 2015) provides a general architecture for sequence-to-sequence prediction problems. This approach has become very popular in many applications where inputs and outputs are sequential, such as MT. In this approach, an input sequence is encoded into an internal representation, and then an output sequence is generated left to right from this representation. Current best practices implement encoder-decoder approaches using RNNs, which handle inputs as a sequence (here a sequence of words), while taking previous words into account. We consider two different architectures, both based on the RNN encoder-decoder approach:

POSTECH We reimplement[1] the neural-based architecture for multi-level prediction by Kim et al. (2017a; Kim et al. (2017b), the best performing system at the WMT 2017 shared task on QE. This architecture is a two-stage end-to-end stacked neural QE model that combines (a) a *predictor* step, an encoder-decoder RNN model to predict words based on their context representations; and (b) an *estimator* step, a bidirectional RNN model to produce quality estimates for words, phrases and sentences based

[1]No code was originally available; our implementation is based on the NMT-Keras framework (Peris, 2017), and the Keras tool (Chollet and others, 2015). The code is publicly available online: `https://github.com/sheffieldnlp/deepQuest`.

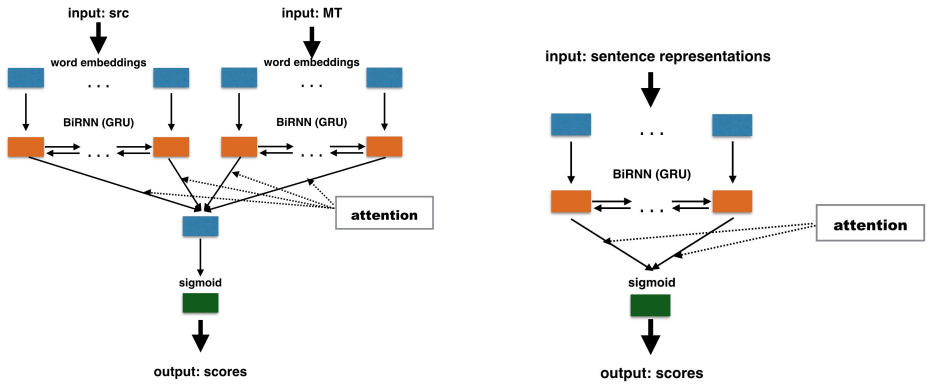

Figure 1: deepQuest framework. Left: the proposed sentence-level QE architecture: hidden states of source and MT encoders are concatenated and an attention mechanism over words is applied. Right: our document-level QE architecture: sentence representations are given to a bi-RNN and an attention mechanism over outputs of this RNN is applied to weight sentences according to their importance to the document.

on representations from the predictor. `POSTECH` requires extensive predictor pre-training to be effective, which means dependence on large parallel data and computational resources.

`BI-RNN` `BI-RNN` uses only two bi-directional RNNs (bi-RNN) as encoders to learn the representation of the (source, MT) sentence pair. A bi-RNN typically calculates a forward sequence of hidden states $(\overrightarrow{h}_1, ..., \overrightarrow{h}_J)$, and a backward sequence of hidden states $(\overleftarrow{h}_1, ..., \overleftarrow{h}_J)$. The hidden states $\overrightarrow{h}_j$ and $\overleftarrow{h}_j$ are concatenated to obtain the resulting representation h_j. In our approach, source and MT bi-RNNs are trained independently, as illustrated in Figure 1 (left side). The two representations are then combined via concatenation. However, sentence-level QE scores are not simple aggregations of word-level representations: they reflect some importance of words within a sentence. Thus, weights should be applied to those representations. Such weighting is provided by the attention mechanism. We apply the following attention function computing a normalized weight for each hidden state h_j:

$$\alpha_j = \frac{\exp(W_a h_j^\top)}{\sum_{k=1}^{J} \exp(W_a h_k^\top)}. \quad (1)$$

The resulting sentence vector is thus a weighted sum of word vectors: $v = \sum_{j=1}^{J} \alpha_j h_j$. A `sigmoid` output layer takes this vector as input and produces real-value quality scores.[2]

3.2 Document-level architecture

Our document-level framework uses a bi-RNN encoder. RNNs have been successfully used for document representation (Lin et al., 2015) and applied to a series of downstream tasks such as topic labeling, summarization, and question answering (Li et al., 2015; Yang et al., 2016).

The document-level quality predictor takes as input a set of sentence-level representations. The last hidden state of the decoder is the summary of an entire sequence. The sum, the maximum, or the average of hidden states for each sentence can then be provided to the output layer. Our assumption is that document-level QE scores are not a simple aggregations of sentence-level QE scores: they should reflect some notion of the importance of sentences within a document. To do so, we use an attention mechanism (Equation 1) to learn weights of different representations (different sentences). The weighted sum of the sentence representations is provided to the `sigmoid` output layer. This `sigmoid` output layer produces real-value predictions for a document, as illustrated in Figure 1 (right side).

[2]Note that Jhaveri et al. (2018) also propose an RNN-based QE architecture. However, following traditional NMT design, the attention mechanism is applied to the source encoder outputs, whereas we apply it to the outputs of both source and MT encoders.

4 Evaluation

In what follows, we first assess the performance of the sentence-level prediction approaches (Section 4.1), then move to the more coarse-grained level of predicting quality for documents (Section 4.2).

4.1 Sentence-level predictions

In this section we analyse the performance of our two sentence-level architectures against official baselines of the WMT 2017 QE shared task. We use the QE dataset in (Specia et al., 2017), which i) is a superset of the official dataset for the WMT 2017 QE task and for which the POSTECH architecture has been developed, allowing us to validate our reimplementation; ii) contains translations from neural MT systems. We also report results using the official evaluation metrics of the shared task: Pearson ρ correlation (primary) and Mean Average Error (MAE) (secondary).

Data The QE dataset contains 28,000 English-German (EN–DE) translations (IT domain) and 18,768 English-Latvian (EN–LV) translations (Life Science domain), produced with either a statistical MT (SMT), or an NMT system.

We randomly split the data for each language pair into training set (25K sentences for EN–DE, 16K for EN–LV), development set (1K sentences) and test (2K sentences) set.[3] In line with the WMT QE campaigns, data labelling was performed as described in (Bojar et al., 2017) using the TERCOM toolkit:[4] for sentence-level QE, with edit distance scores (HTER) used as labels.

To train POSTECH's predictor:[5] we used the Europarl corpus (Koehn, 2005) ($\approx$ 2M sentences; the version provided by Tiedemann (2012)) for EN–DE, and the parallel data of the WMT 2017 News translation task ($\approx$ 2M sentences) for EN–LV. Each experiment was run five times on the same split to estimate the stability of the model.

Baseline system: We reproduced the WMT 2017 baselines as described in (Bojar et al., 2017). For the extraction of the sentence-level features for EN–DE, we used the additional resources provided by the WMT 2017 QE shared task. For EN–LV, the corresponding resources were created using the data provided for the WMT 2017 News translation task, and the EMEA corpus (Tiedemann, 2009).

Implementation details: We implemented the sentence-level architectures using the Keras toolkit with Gated Recurrent Units (GRUs) (Cho et al., 2014) as RNNs, and the following hyperparameters: word embedding dimensionality = 300, vocabulary size = 30K, size of the hidden units of the encoder = 50. The model was trained to minimize the mean squared error loss using the Adadelta optimizer (Zeiler, 2012).

Results The results of our experiments are reported in Table 1. In general, different runs of the models do not lead to much variation in the performance. For EN–DE, a first observation is the major improvement in NMT quality compared to SMT (HTER = 0.09 vs. HTER = 0.24), which results in highly imbalanced NMT datasets (for EN–DE, $\approx$54% of all the sentences have 0 HTER vs. $\approx$13% for SMT). For EN–LV, the quality of NMT, limited by the amount of training data, is worse than SMT (HTER = 0.15 vs. HTER = 0.23).

These results show that the performance of baseline methods depends on the quality of translations rather than on the type of system that produced them. The baseline EN–DE methods achieve ρ scores that are 60% worse for NMT translations, as compared to SMT translation. For EN–LV, the behavior is the opposite: the performance is 45% better for NMT translations, as compared to SMT translations.

The performance of POSTECH is on average 40% higher for SMT than for NMT (e.g. for EN–LV, ρ=0.39 for SMT vs. ρ=0.24 for NMT). For EN–DE systems, this can be attributed to the data imbalance. For EN–LV systems, this could be because the neural QE system has difficulty to handle the many very

[3] As only part of the EN–DE SMT data was used for the WMT 2017 QE task, we could not use the task's official split.

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

[5] Hyperparameters size of hidden units of the word predictor = 500, word embedding dimensionality = 300, vocabulary size = 30K, QE vector size = 75.

long sentences produced by the EN–LV NMT ($\sigma^2 = 92$ of the distribution of the sentence length values for NMT vs. $\sigma^2 = 73$ for SMT).

For NMT the difference in variance of predicted HTER scores between language pairs is relatively high (as reflected in MAE differences between language pairs for neural methods, e.g., for POSTECH $\Delta = 0.084$), whereas for SMT this difference is lower (for POSTECH MAE $\Delta = 0.002$).

In general, without pre-training POSTECH does not perform better than the baseline methods (e.g. for EN–DE SMT, ρ=0.313 for the baseline vs. ρ=0.324 for POSTECH) and is systematically outperformed by BI-RNN (average $\Delta\rho$=0.06). This difference is statistically significant for both language pairs.[6] We believe that BI-RNN is able to better capture the fluency of NMT by encoding it directly as a sequence rather than assessing it word for word as POSTECH.

	EN–DE		**EN–LV**	
model	ρ	MAE	ρ	MAE
Baseline				
SMT	0.313 ±0.0	0.147 ±0.0	0.100 ±0.0	0.056 ±0.0
NMT	0.130 ±0.0	0.171 ±0.0	**0.318** ±0.0	0.070 ±0.0
POSTECH (no pre-training)				
SMT	0.324 ±0.015	0.146 ±0.002	0.294 ±0.015	0.136 ±0.002
NMT	0.153 ±0.023	0.103 ±0.001	0.240 ±0.010	0.205 ±0.008
POSTECH				
SMT	**0.481**±0.009	0.131 ±0.002	**0.390** ±0.016	0.129±0.005
NMT	**0.318** ±0.014	0.092 ±0.002	0.240 ±0.004	0.176 ±0.004
BI-RNN				
SMT	0.363 ±0.010	0.142 ±0.002	0.357 ±0.004	0.133 ±0.004
NMT	0.311±0.004	0.090 ±0.003	0.231 ±0.012	0.183 ±0.004

Table 1: Performance on sentence-level predictions for the baseline with QuEst, the POSTECH architecture and our BI-RNN architecture for EN–DE and EN–LV (average and error margins over five runs). We highlight the best performing systems for a dataset.

4.2 Document-level predictions

As introduced above, our framework relies on representations at sentence level to produce its predictions at document level. Therefore, we experiment with sentence-level representations from either the POSTECH or our BI-RNN predictor.

The document-level labels we predict are variants of BLEU (Papineni et al., 2002): (i) document-level BLEU,[7] (ii) the weighted average of sentence-level BLEU for all sentences in the document, where the weights correspond to the reference lengths:

$$\text{wBLEU}_d = \frac{\sum_{i=1}^{D} \text{len}(R_i)\text{BLEU}_i}{\sum_{i=1}^{D} \text{len}(R_i)},$$

where BLEU_i is the BLEU score of sentence i, and D is the size of the document in sentences,[8] and (iii), a variant of WBLEU by weighting each sentence by its TFIDF score computed with regard to its aligned reference (TBLEU). The numerator in the WBLEU equation is therefore replaced by: $\sum_{i=1}^{D} \text{TFIDF}_i\text{BLEU}_i$. Here, for each news document, we learn a TFIDF model on its reference in the target language, and compute the TFIDF score for each translated sentence, based on that model.

[6] We performed Kolmogorov–Smirnov test for not normally distributed data.

[7] We compute BLEU scores with the NLP toolkit NLTK (Bird and Loper, 2004). For scoring documents, we used the corpus_bleu() function. For sentence-level scores we used the sentence_bleu() function with smoothing method 7.

[8] According to Chen and Cherry (2014), WBLEU achieves a better correlation with human judgement than the original IBM corpus-level BLEU.

Following Turchi et al. (2012), our intuition is that the document-level score should reflect the overall translation quality at sentence level, weighted by how important each individual sentence (important sentences have important words) is in that document.

Data: We gathered all submissions at the WMT News shared tasks for various years. This is a task where each participating system is required to translate a set of news documents.[9] This results in a large set of language pairs, as well as a wide range of different translation quality levels. We collected system submissions from WMT 2008 to 2017, for four language pairs: German-English (DE–EN, 14,640 documents for 2008-2017, excluding 2010),[10] and English–Spanish (EN–ES, 6,733 documents for 2008-2013, excluding 2010[10]), English–French (EN–FR, 11,537 documents for 2008-2014) and English-Russian (EN–RU, 6,996 documents for 2013-2017). For each language pair, we consider either the full set of system submissions, or a filtered version of it (FILT), composed by only both the best and the worst performing systems for each year. The filtering was done based on the overall BLEU score achieved by each system, as reported on `matrix.statmt.org`. Our intuition is that by considering only the extreme quality levels we would make our data, while smaller, easier to discriminate.[11] We note that for all language pairs, MT systems include a variety of approaches, from rules-based MT, to SMT, hybrid approaches, and – from 2016 – NMT approaches. The filtered variants include at least one NMT approach for the language pairs in 2016/2017 (i.e. DE–EN and EN–RU). To support the reproduction of our work, as well as the development of new models for document-level QE, we release this dataset.[12]

Taking the heterogeneity of the data into account (composed of outputs of different systems and runs), to evaluate our models we perform 5-fold cross validation. For each language pair we shuffle the data, and for each fold we split it per year into train, development and test sets, as follows: for the FILT dataset, 10% of the documents per year were randomly selected for the development set, an another 10% for the test set; for ALL, the train data in the quantity equal to the FILT train data was selected randomly (development and test data were fixed, since ALL contains FILT). We present the averaged results. and use Kolmogorov–Smirnov test for not normally distributed data to test significance. As an example, statistics on one of the splits are shown in Table 2. Note that to avoid computation precision issues, we multiply TBLEU scores by 10.

As `POSTECH` training is very expensive, for the contrastive experiments we use only DE–EN and EN–ES. For the experiments with `BI-RNN`, we use all language pairs.

Baseline system: We reproduced the WMT 2016 document-level baseline as described in (Bojar et al., 2016b). We extract 17 black-box features using `QuEst++` (Specia et al., 2015) and train a document-level QE system using the Support Vector Regression (SVR) algorithm available in `scikit-learn` (Pedregosa et al., 2011). The language resources were created using the News Commentary and Europarl corpora as provided by WMT campaigns for the corresponding languages ($\approx$ 2M lines per language pair). These corpora were also used to train `POSTECH` predictors.[13]

For `BI-RNN`, we followed the implementation details as described in Section 4.1. To optimize the usage of computational resources, in each experiment we fixed the size of a document to the upper quartile of the distribution of document length values (in sentences). Shorter documents were extended with dummy sentences to fit to this length, which is a common practice in the field (Hewlett et al., 2017).

Results Results of our experiments are reported in Tables 3 (baseline) and 4 (neural approaches). For both `POSTECH` and `BI-RNN`, we use only the last hidden states of the document decoder (`Last`) – a configuration that has been chosen empirically as the best performing among the configurations without attention, or the vector sum weighted by the attention mechanism (`Att`) as input to the output layer.

[9] We considered using the document-level QE dataset in (Graham et al., 2017), however, the small number of documents (62) and language pairs (only one) made this resource less appealing for this work.

[10] Individual submissions are not available but system combinations only

[11] In (FILT) we have also discarded submissions for years 2008 and 2009, since official system-level scores are not available.

[12] `https://github.com/fredblain/docQE`

[13] `http://www.statmt.org/wmt18/translation-task.html;http://www.statmt.org/wmt13/translation-task.html`

set	# docs	av # sent	BLEU	wBLEU	tBLEU
DE–EN					
FILT	1147	26	0.529	0.316	0.457
ALL	1147	26	0.530	0.316	0.461
dev	140	26	0.527	0.319	0.454
test	140	26	0.521	0.314	0.459
EN–ES					
FILT	420	35	0.516	0.324	0.423
ALL	420	34	0.529	0.324	0.420
dev	51	34	0.504	0.323	0.424
test	51	34	0.533	0.324	0.426
EN–FR					
FILT	894	26	0.442	0.318	0.445
ALL	894	27	0.500	0.320	0.446
dev	109	27	0.464	0.320	0.454
test	109	24	0.475	0.320	0.440
EN–RU					
FILT	1052	22	0.591	0.329	0.561
ALL	1052	23	0.587	0.329	0.562
dev	130	22	0.585	0.330	0.561
test	130	24	0.581	0.330	0.551

Table 2: Statistics of the document-level dataset gathered from all submissions at the WMT News translation shared task. The first column presents the dataset considered, while second and third columns report the number of news documents in a dataset and the average number of sentences per document. The last three columns report respectively the average BLEU, average weighted BLEU (wBLEU) and the average variant of weighted BLEU (tBLEU) we propose in that set.

	DE–EN		**EN–ES**	
	ρ	MAE	ρ	MAE
	FILT			
BLEU	**0.065**	0.477	0.024	0.064
wBLEU	**0.177**	0.010	0.032	0.008
tBLEU	**0.043**	0.045	0.046	0.047
	ALL			
BLEU	0.044	0.973	**0.143**	0.063
wBLEU	0.033	0.010	**0.051**	0.007
tBLEU	0.019	0.046	**0.050**	0.050

Table 3: Baseline document-level score prediction results for DE–EN and EN-ES. We highlight the best performing systems for a dataset.

A first observation is that the performance of our neural approach varies significantly for different quality labels: the best performance is systematically observed for tBLEU, the worst – for BLEU (e.g. for DE–EN neural models, ρ=0.69 vs. ρ=0.39, on average respectively). This is not true for the baselines where the best performance is observed for other scores depending on the training data used. We attribute the high prediction performance for tBLEU to the fact that our architecture builds document-level representation from sentence-level representations, which in turn depend on word representations. The tBLEU reflects this hierarchy in the most consistent way as those document-level scores depend directly on semantic importance of words they contain. BLEU, on the other hand, depends on n-gram translation quality. MAE is in general the lowest for wBLEU with the lowest variance.

	DE–EN				EN–ES			
score	Last		Att		Last		Att	
	ρ	MAE	ρ	MAE	ρ	MAE	ρ	MAE
	FILT							
	POSTECH							
BLEU	0.122	0.064	0.170	0.060	0.472	0.078	0.548	0.070
WBLEU	0.330	0.010	0.511	0.007	0.317	0.015	0.300	0.032
TBLEU	0.632	0.035	0.744	0.088	0.739	0.030	0.854	0.020
	BI-RNN							
BLEU	0.213	0.056	0.157	0.050	0.487	0.084	**0.568**	0.070
WBLEU	0.413	0.008	0.590	0.007	0.512	0.072	0.407	0.025
TBLEU	0.770	0.031	0.814	0.029	0.898	0.020	0.903	0.020
	ALL							
	POSTECH							
BLEU	0.306	4.735	0.344	4.730	0.476	0.075	0.365	0.074
WBLEU	0.536	0.007	0.544	0.007	0.491	0.008	0.345	0.028
TBLEU	0.742	0.030	0.807	0.027	0.820	0.025	0.895	0.019
	BI-RNN							
BLEU	0.317	4.724	**0.363**	3.816	0.471	0.073	0.439	0.191
WBLEU	**0.660**	0.007	0.650	0.006	0.617	0.012	**0.655**	0.016
TBLEU	0.854	0.024	**0.889**	0.022	0.927	0.017	**0.941**	0.017

	EN–FR				EN–RU			
score	Last		Att		Last		Att	
	ρ	MAE	ρ	MAE	ρ	MAE	ρ	MAE
	FILT							
	BI-RNN							
BLEU	0.517	0.125	0.687	0.103	0.379	0.087	0.377	0.110
WBLEU	0.425	0.010	**0.504**	0.009	0.648	0.006	0.575	0.005
TBLEU	0.827	0.031	0.815	0.032	0.826	0.034	0.849	0.036
	ALL							
	BI-RNN							
BLEU	0.559	0.103	**0.723**	0.086	0.402	0.087	**0.418**	0.090
WBLEU	0.469	0.008	0.426	0.008	**0.726**	0.005	0.573	0.006
TBLEU	0.838	0.023	**0.844**	0.022	0.866	0.029	**0.876**	0.028

Table 4: Document-level score prediction results for neural approaches for DE–EN, EN–ES, EN–FR, and EN–RU. Last refers to the results after we take the last hidden state of the document-level encoder as input to the output layer; Att – with an attention mechanism. We highlight the best performing systems for a dataset.

The baseline yields poor performance (e.g., for BLEU ρ=0.07 on average across configurations), whereas BI-RNN systematically outperforms POSTECH for all three prediction tasks and across configurations (e.g., for DE–EN $\Delta\rho$=0.08). For DE–EN, this difference is statistically significant, while for EN–ES it is not. We believe this can be explained by lower stability of classifiers trained on the smaller EN–ES dataset.

The best performance improvement is observed for WBLEU, which we believe is because BI-RNN is less "focused" on word-level predictions (e.g., for DE–EN $\Delta\rho$=0.10). Additionally, BI-RNN is about 40 times faster to train than POSTECH.[14]

[14] BI-RNN takes around 20 minutes to train on a 12G GeForce TITAN X NVIDIA GPU with batch size = 10. Pre-training a POSTECH model in the same conditions with around 2M lines of parallel data takes around 12 hours plus the training of the

The contribution of the attention mechanism also depends on the type of quality label, but it is not statistically significant. It can be particularly beneficial for BLEU (for instance, for BI-RNN DE–EN across configurations, $\Delta\rho$=0.10 on average), but particularly harmful for WBLEU (for instance, for BI-RNN EN–RU, a decrease of ρ=0.12 is observed). This can be explained by the influence of variable reference lengths and hence the difficulty to find optimal weights.

The training data filtering procedure is not beneficial for the performance. This procedure is particularly harmful for DE–EN BLEU and WBLEU scores (for BI-RNN, average $\Delta\rho$=0.11), which is the most well represented language pair across WMT years. Thus, it may be the case that the random ALL selection contains more useful data.

Pre-training the predictor for POSTECH is essential for this architecture; for EN–ES BLEU Last, for example, a decrease of up to $\Delta\rho$=0.3 is observed without pre-training.

As for the difficulty of prediction for different language pairs: DE–EN and EN–RU BLEU prediction seems to be the most challenging (for BI-RNN, on average ρ=0.26 and ρ=0.40, respectively). This could be explained by the traditionally lower MT quality for those systems, involving significant word order differences for these language pairs.

5 Conclusions

We have proposed a new approach for neural-based document-level QE that is able to generalize any neural sentence-level architecture to the level of documents. This approach is part of our new framework for QE, named deepQuest, that reimplements the state of the art neural-based architecture to date for sentence-level quality prediction – the POSTECH approach – as well as our light-weight neural architecture relying on bi-directional RNNs. Our experiments have shown that latter outperforms POSTECH when used to predict document-level quality estimates for a range of quality scores and is 40 times faster to train. We also have reported a study of the performance of state of the art QE approaches on NMT output. Neural QE solutions are more efficient for imbalanced QE data, especially high-quality NMT. To our knowledge, this is the first time that results of QE on a large range of NMT data are reported.

In the future, we plan to reproduce our study for other document-level QE scores and text types. We also plan to adapt our light-weight neural architecture for other QE levels (e.g., phrase, paragraph levels). To support future work on these and other directions, we have made deepQuest open-source and freely available: https://github.com/sheffieldnlp/deepQuest.

Acknowledgements

The development of deepQuest received funding from the European Association for Machine Translation and the Amazon Academic Research Awards program. The first author worked on this paper during a research stay at the University of Sheffield.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Steven Bird and Edward Loper. 2004. NLTK: the natural language toolkit. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Interactive poster and demonstration sessions (ACL)*, page 31.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58.

document-level system on average 2 additional hours. BI-RNN can also be trained on a CPU in about 3 hours.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016a. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation (WMT)*, pages 131–198.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016b. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation (WMT)*, pages 131–198.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation (WMT)*, pages 169–214.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT)*, pages 362–367.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

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

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Yvette Graham, Qingsong Ma, Timothy Baldwin, Qun Liu, Carla Parra, and Carolina Scarton. 2017. Improving evaluation of document-level machine translation quality estimation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 356–361, Valencia, Spain, April. Association for Computational Linguistics.

Daniel Hewlett, Llion Jones, Alexandre Lacoste, and izzeddin gur. 2017. Accurate supervised and semi-supervised machine reading for long documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2011–2020.

Nisarg Jhaveri, Manish Gupta, and Vasudeva Varman. 2018. Translation quality estimation for indian languages. In *Proceedings of th 21st International Conference of the European Association for Machine Translation (EAMT)*.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017a. Predictor-Estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 17(1):3:1–3:22, September.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017b. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation (WMT)*, pages 562–568, September.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the X Machine Translation Summit*, pages 79–86.

Jiwei Li, Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 1106–1115.

Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. 2015. Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 899–907.

André Martins, Marcin Junczys-Dowmunt, Fabio Kepler, Ramón Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017a. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics*, 5:205–218.

André F. T. Martins, Fabio Kepler, and Jose Monteiro. 2017b. Unbabel's participation in the wmt17 translation quality estimation shared task. In *Proceedings of the Second Conference on Machine Translation (WMT)*, pages 569–574, September.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Raj Nath Patel and Sasikumar M. 2016. Translation quality estimation using recurrent neural network. In *Proceedings of the First Conference on Machine Translation (WMT)*, pages 819–824, August.

Fabian Pedregosa, Gaël Varoquaux, Vincent Michel Alexandre Gramfort, Bertrand Thirion, Olivier Grisel, Peter Prettenhofer Mathieu Blondel, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Álvaro Peris. 2017. NMT-Keras. `https://github.com/lvapeab/nmt-keras`. GitHub repository.

Carl Edward Rasmussen and Christopher K. I. Williams. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. MIT Press.

Matīss Rikters and Mark Fishel. 2017. Confidence through attention. In *Proceedings of the 16th Machine Translation Summit (MT Summit XVI)*, pages 299–312.

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 (WMT)*, pages 831–837.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *Proceedings of 13th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 28–37.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with QuEst++. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*, pages 115–120.

Lucia Specia, Kim Harris, Frédéric Blain, Aljoscha Burchardt, Viviven Mackentanz, Inguna Skadiņa, Matteo Negri, and Marco Turchi. 2017. Translation quality and productivity: A study on rich morphology languages. In *Proceedings of the 16th Machine Translation Summit (MT Summit XVI)*, pages 282–299.

Ilya Sutskever, Oriol Vinyals, and Quoc V. V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume V, pages 237–248.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*.

Marco Turchi, Lucia Specia, and Josef Steinberger. 2012. Relevance ranking for translated texts. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 153–160.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 1480–1489.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.