SHEF-LIUM-NN: Sentence-level Quality Estimation with Neural Network Features

Kashif Shah[§], **Fethi Bougares**[†], **Loïc Barrault**[†] **Lucia Specia**[§]
§Department of Computer Science, University of Sheffield, UK
{kashif.shah,l.specia}@sheffield.ac.uk

†LIUM, University of Le Mans, France
{fethi.bougares,loïc.barrault}@lium.univ-lemans.fr

Abstract

This paper describes our systems for Task 1 of the WMT16 Shared Task on Quality Estimation. Our submissions use (i) a continuous space language model (CSLM) to extract sentence embeddings and cross-entropy scores, (ii) a neural network machine translation (NMT) model, (iii) a set of QuEst features, and (iv) a combination of features produced by QuEst and with CSLM and NMT. Our primary submission achieved third place in the scoring task and second place in the ranking task. Another interesting finding is the good performance obtained from using as features only CSLM sentence embeddings, which are learned in an unsupervised fashion without any additional hand-crafted features.

1 Introduction

Quality Estimation (QE) aims at measuring the quality of the output of Machine Translation (MT) systems without reference translations. Generally, QE is addressed with various features indicating fluency, adequacy and complexity of the source and translation texts. Such features are used along with Machine Learning methods in order to learn prediction models.

Features play a key role in QE. A wide range of features from the source segments and their translations, often processed using external resources and tools, have been proposed. These go from simple, language-independent features, to advanced, linguistically motivated features. They include features that rely on information from the MT system that generated the translations, and features that are oblivious to the way translations were produced. This leads to a potential bottleneck: feature engineering can be time consuming, particularly because the impact of features vary across datasets and language pairs. Also, most features in the literature are extracted from segment pairs in isolation, ignoring contextual clues from other segments in the text. The focus of our contributions this year is to explore a new set of features which are language-independent, require minimal resources, and can be extracted in unsupervised ways with the use of neural networks.

Word embeddings have shown their potential in modelling long distance dependencies in data, including syntactic and semantic information. For instance, neural network language models (Bengio et al., 2003) have been successfully explored in many problems including Automatic Speech Recognition (Schwenk and Gauvain, 2005; Schwenk, 2007) and Machine Translation (Schwenk, 2012).

In this paper, we extend our previous work (Shah et al., 2015a; Shah et al., 2015b) to investigate the use of sentence embeddings extracted from a neural network language model along with cross entropy scores as features for QE. We also investigate the use of a neural machine translation model to extract the log likelihood of sentences as QE features. The features extracted from such resources are used in isolation or combined with hand-crafted features from QuEst to learn prediction models.

2 Continuous Space Language Model Features

Neural networks model non-linear relationships between the input features and target outputs. They often outperform other techniques in complex machine learning tasks. The inputs to the neural network language model used here (called Continuous Space Language Model (CSLM)) are

Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers, pages 838–842,
Berlin, Germany, August 11-12, 2016.

the h_j context words of the prediction: $h_j = w_{j-n+1}, ..., w_{j-2}, w_{j-1}$, and the outputs are the posterior probabilities of all words of the vocabulary: $P(w_j|h_j)\ \forall i \in [1, N]$ where N is the vocabulary size. A CSLM encodes inputs using the so called one-hot coding, i.e., the ith word in the vocabulary is coded by setting all elements to 0 except the ith element. Due to the large size of the output layer (vocabulary size), the computational complexity of a basic neural network language model is very high. Schwenk (2012) proposed an implementation of the neural network with efficient algorithms to reduce the computational complexity and speed up the processing using a subset of the entire vocabulary called *short list*.

As compared to shallow neural networks, deep neural networks can use more hidden layers and have been shown to perform better (Schwenk et al., 2014). In all CSLM experiments described in this paper, we use 40-gram deep neural networks with four hidden layers: a first layer for the word projection (320 units for each context word) and three hidden layers of 1024 units for the probability estimation. At the output layer, we use a $softmax$ activation function applied to a *short list* of the 32k most frequent words. The probabilities of the out of the *short list* words are obtained using a standard back-off n-gram language model. The training of the neural network is done by the standard back-propagation algorithm and outputs are the posterior probabilities. The parameters of the models are optimised on a held out development set. Our CSLM models were trained with the CSLM toolkit [1] and used to extract the following features:

- source sentence cross-entropy
- source sentence embeddings
- translation output cross-entropy
- translation output embeddings.

Table 1, reports detailed statistics on the monolingual data used to train the back-off LM and CSLM. The training dataset consists of WMT16 translation task monolingual corpora with the Moore-Lewis data selection method (Moore and Lewis, 2010) to select the CSLM training data with respect to the task's development set. The CSLM models are tuned using the WMT16 Quality Estimation development corpus.

Lang.	Train	Dev	4-g LM px	CSLM px
en	84G	17.8 k	61.30	50.69
de	79G	19.7 k	64.99	54.45

Table 1: Training and dev datasets size (in number of tokens) and models perplexity (px).

3 Neural Machine Translation Features

In addition to the monolingual features learned using the neural network language model, we experiment with bilingual features derived from a neural machine translation system (NMT). Our NMT system is developed based on a framework inspired from the dl4mt-material project[2]. The system is an end-to-end sequence to sequence model tuned to minimise the negative log-likelihood using a stochastic gradient descent. In our experiments we trained two NMT systems (EN $\leftrightarrow$ DE) with an attention mechanism similar to the one described in (Bahdanau et al., 2014).

Let X and Y be a source sentence of length T_x and a target sentence of length T_y respectively:

$$X = (x_1, x_2, ..., x_{T_x}) \quad (1)$$

$$Y = (y_1, y_2, ..., y_{T_y}) \quad (2)$$

Each source and target word is represented with a randomly initialised embedding vector of size E_s and E_t respectively. A bidirectional recurrent encoder reads an input sequence X in forward and backward directions to produce two sets of hidden states. At the end of the encoding step, we obtain a bidirectional annotation vector h_t for each source position by concatenating the forward and backward annotations:

$$h_t = \begin{bmatrix} \vec{h_t} \\ \overleftarrow{h_t} \end{bmatrix} \quad (3)$$

A Gated Recurrent Unit (GRU) (Chung et al., 2014) is used for the encoder and decoder. They have 1000 hidden units each, leading to an annotation vector $h_t \in \mathbb{R}^{2000}$.

The attention mechanism, implemented as a simple fully-connected feed-forward neural network, accepts the hidden state h_t of the decoder's recurrent layer and one input annotation at a time,

[1] http://www-lium.univ-lemans.fr/cslm/

[2] github.com/kyunghyuncho/dl4mt-material

to produce the attention coefficients. A softmax activation is applied on those attention coefficients to obtain the attention weights used to generate the weighted annotation vector for time t.

Both NMT systems are trained with WMT16 Quality Estimation English-German datasets (we used post-editions on the German side) and tuned on the official development set. Table 2 reports the statistics of NMT training data and BLEU scores on the QE development set.

Trans. Direction	Train	Dev	BLEU
DE-to-EN	21k-20k	17.8 k	35.38
EN-to-DE	20k-21k	19.7 k	37.51

Table 2: Training and development datasets sizes (number of tokens) and development set BLEU scores.

4 Experiments

In what follows we present our experiments on the WMT16 QE Task 1 with CSLM and NMT features.

4.1 Dataset

Task 1's English-German dataset consists respectively of a training set and development set with $12,000$ and $1,000$ source segments, their machine translations, the post-editions of the latter, and the edit distance scores between the MT and its post-edited version (HTER). The test set consists of $2,000$ English-German source-MT pairs. Each of the translations was post-edited by professional translators, and HTER labels were computed using the TER tool (settings: tokenised, case insensitive, exact matching only, with scores capped to 1).

4.2 Features

We extracted the following features:

- **QuEst**: 79 black-box features using the QuEst framework (Specia et al., 2013; Shah et al., 2013a) as described in Shah et al. (2013b). The full set of features can be found on `http://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox`.
- **CSLM**$_{ce}$: A cross-entropy feature for each source and target sentence using CSLM as described in Section 2.
- **NMT**$_{ll}$: A log likelihood feature for each source and target sentence using NMT as described in Section 3.
- CSLM$_{emb}$: Sentence features extracted by taking the mean of 320-dimension word vectors trained using CSLM for both source and target. We also experimented with taking the min or the max of the embeddings, but empirically it was found that the mean performs better. Therefore, all our results are reported using the mean of word embeddings.

4.3 Learning algorithm

We use the Support Vector Machines implementation in the `scikit-learn` toolkit (Pedregosa et al., 2011) to perform regression (SVR) on each feature set with either RBF kernels and parameters optimised using grid search.

To evaluate the prediction models we use all evaluation metrics in the task: Pearson's correlation r, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Spearman's correlation ρ and Delta Average (DeltaAvg).

4.4 Results

We trained various models with different feature sets and algorithms and evaluated the performance of these models on the official development set. The results are shown in Table 3. Based on these findings, as official submissions for Task 1, we submitted two systems:

- SHEF-SVM-CSLM$_{ce}$-NMT$_{ll}$-CSLM$_{both-emb}$
- SHEF-SVM-QuEst-CSLM$_{ce}$-NMT$_{ll}$-CSLM$_{both-emb}$

These systems contain all of our CSLM and NMT features either with or without QuEst: 719 and 644 features in total, respectively. We named them SVM-NN-both-emb and SVM-NN-both-emb-QuEst in the official submissions. The official results are shown in Table 4. Our systems show promising performance across all of the metrics used for evaluation in both scoring and ranking task variants. Our best system was ranked:

- Third place in the scoring task variant according to Pearson r (official scoring metric), and second place according MAE and RMSE.
- Second place in the ranking task variant according to Spearman ρ (official ranking metric) and first place according to DeltaAvg.

System.	# of Feats.	MAE	RMSE	Pearson r
Baseline (SVM)	17	13.97	19.65	0.359
SHEF-SVM-QuEst	79	13.94	19.71	0.386
SHEF-SVM-QuEst-CSLM$_{ce}$-NMT$_{ll}$	83	14.27	19.92	0.460
SHEF-SVM-CSLM$_{src-emb}$	320	13.97	18.87	0.416
SHEF-SVM-CSLM$_{tgt-emb}$	320	13.70	18.60	0.422
SHEF-SVM-CSLM$_{both-emb}$	640	13.74	18.10	0.425
SHEF-SVM-CSLM$_{ce}$-NMT$_{ll}$-CSLM$_{both-emb}$	644	13.48	17.94	**0.500**
SHEF-SVM-QuEst-CSLM$_{ce}$-NMT$_{ll}$-CSLM$_{tgt-emb}$	383	13.49	17.99	0.500
SHEF-SVM-QuEst-CSLM$_{ce}$-NMT$_{ll}$-CSLM$_{both-emb}$	719	13.46	17.92	**0.501**

Table 3: Results on the development set of Task 1. Systems in bold are used as official submissions.

System.	MAE	RMSE	Pearson r	DeltaAvg	Spearman ρ
Baseline	13.53	18.39	0.351	62.981	0.390
SVM-NN-both-emb	12.97[3]	17.33[3]	0.430[5]	78.86[1]	0.452[2]
SVM-NN-both-emb-QuEst	12.88[2]	17.03[2]	0.451[3]	81.30[1]	0.474[2]

Table 4: Official results on the test set of Task 1. The superscript shows the overall ranking of the system against various official evaluation metrics.

Some of the interesting findings are:

- The mean of word embeddings extracted for each sentence performs much better than the max or min.
- Sentence features extracted from CSLM embeddings bring the largest improvements.
- Target embeddings produce better predictions than source embeddings, which is inline with our previous findings (Shah et al., 2015b).
- CSLM cross entropy and NMT log likelihood features bring further improvements on top of embedding features.
- QuEst features bring improvements whenever added to either CSLM embeddings or cross entropy and NMT likelihood features.
- Neural Network features alone perform very well. This is a very encouraging finding since for many language pairs it can be difficult to find appropriate resources to extract handcrafted features.

5 Conclusions

In this paper we have explored novel features for translation Quality Estimation which are obtained with the use of Neural Networks. When added to QuEst standard feature sets for the WMT16 QE Task 1, the CSLM sentence embedding features along with cross entropy and NMT likelihood led to large improvements in prediction. Moreover, CSLM and NMT features alone performed very well. Combining all CSLM and NMT features with the ones produced by QuEst improved the performance and led to very competitive systems according to the task's official results.

In the future work, we plan to explore bilingual embeddings extracted from our NMT models. Compared to the CSLM embeddings, NMT models generate embeddings (with the bidirectional Neural Network as presented in Section 3) of the whole sentence with a focus on the current word. In addition, we plan to train a Neural Network model to directly predict the QE scores.

Acknowledgements

This work was supported by the QT21 (H2020 No. 645452, Lucia Specia), Cracker (H2020 No. 645357, Kashif Shah) and the Chist-ERA M2CR[3] (Fethi Bougares and Loïc Barrault) projects.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

[3] m2cr.univ-lemans.fr

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the 48th ACL*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*.

Holger Schwenk and Jean-Luc Gauvain. 2005. Training neural network language models on very large corpora. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*.

Holger Schwenk, Fethi Bougares, and Loic Barrault. 2014. Efficient training strategies for deep neural network language models. *Proceedings of NIPS*.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech & Language*.

Holger Schwenk. 2012. Continuous space translation models for phrase-based statistical machine translation. In *Proceedings of COLING*.

Kashif Shah, Eleftherios Avramidis, Ergun Biçicic, and Lucia Specia. 2013a. Quest - design, implementation and extensions of a framework for machine translation quality estimation. *Prague Bull. Math. Linguistics*.

Kashif Shah, Trevor Cohn, and Lucia Specia. 2013b. An investigation on the effectiveness of features for translation quality estimation. In *Proceedings of the Machine Translation Summit*.

Kashif Shah, Varvara Logacheva, Gustavo Paetzold, Frédéric Blain, Daniel Beck, Fethi Bougares, and Lucia Specia. 2015a. Shef-nn: Translation quality estimation with neural networks. In *Tenth Workshop on Statistical Machine Translation*, pages 338–343, Lisboa, Portugal.

Kashif Shah, Raymond W.M. Ng, Fethi Bougares, and Lucia Specia. 2015b. Investigating continuous space language models for machine translation quality estimation. In *Conference on Empirical Methods in Natural Language Processing*, EMNLP, Lisboa, Portugal.

Lucia Specia, Kashif Shah, José G. C. de Souza, and Trevor Cohn. 2013. QuEst - A translation quality estimation framework. In *Proceedings of 51st ACL*.