UHH Submission to the WMT17 Quality Estimation Shared Task

Melania Duma and Wolfgang Menzel
University of Hamburg
Natural Language Systems Division
{duma, menzel}@informatik.uni-hamburg.de

Abstract

The field of Quality Estimation (QE) has the goal to provide automatic methods for the evaluation of Machine Translation (MT), that do not require reference translations in their computation. We present our submission to the sentence level WMT17 Quality Estimation Shared Task. It combines tree and sequence kernels for predicting the post-editing effort of the target sentence. The kernels exploit both the source and target sentences, but also a back-translation of the candidate translation. The evaluation results show that the kernel approach combined with the baseline features brings substantial improvement over the baseline system.

1 Introduction

The evaluation of Machine Translation (MT) output is a sub-field of MT research that has experienced a great amount of interest in the past years. The process of MT evaluation involves three factors: an input segment in a source language, the candidate translation (also known as target sentence) which represents the output of a MT system when translating from the source language to the target language and a reference translation in the target language. The assessment of MT quality can be divided into two categories depending on whether it requires the presence of a reference translation or not. The reference-based evaluation scores the candidate translation by comparing it to the reference translation.

On the other hand, the reference-free evaluation, also known as quality estimation (QE), predicts the quality of a candidate translation based solely on the information contained in the source and target sentences. QE can be performed at different levels of granularity: word, sentence or phrase and it involves classifying, ranking or predicting scores for the candidate translations. A sentence-level QE system is conventionally constructed based on a set of features encoding the information contained in the source and target sentences, which are used for learning a prediction model. The features employed for this task can be of different types, like surface features, language model features or linguistic features. The positive influence of syntactic features on the performance of QE systems has been extensively studied, including in Rubino et al. (2012), Avramidis (2012) or more recently in Kozlova et al. (2016). However, the process of identifying the best performing set of features, is a task that is both expensive and requires a considerable amount of engineering effort (Hardmeier, 2011). On the other hand, kernel methods do not require the explicit definition of the features, and rely on the scalar product between vectors for capturing the similarity shared by the sentence pairs.

In this paper we present our submission to the WMT17 Shared Task on sentence level Quality Estimation, that makes use of sequence and tree kernels in predicting a continuous score representing the post-editing effort for the target sentence. The novel contribution of our system is the combination of different types of kernels. Moreover, we use a back-translation of the target sentence into the source language as an additional data representation to be exploited by the kernels, together with the usual source and target sentences representations. Furthermore, we construct additional explicit features by applying the kernel functions directly on the pair of source and back-translation sentences, a method that to our knowledge has not been used before. The evaluation performed demonstrates that the combination of the kernel approach and the baseline together with the newly

Proceedings of the Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 556–561
Copenhagen, Denmark, September 711, 2017.

introduced feature vectors brings consistent improvement over the baseline system.

This paper is organized as follows. The related work is presented in Section 2, while the methods employed and the implementation are described in Section 3. The experimental setup and the evaluation results are introduced in Section 4, while the last section summarizes our findings and presents future work ideas.

2 Related work

Kernel functions have been used in a variety of NLP tasks, including Textual Similarity (e.g. (Severyn et al., 2013)), Information Extraction (e.g. (Culotta and Sorensen, 2004)), Semantic Role Labeling (e.g. (Moschitti et al., 2008)) or Textual Entailment (e.g. (Wang and Neumann, 2007)).

An approach for QE based on syntactic tree kernels is introduced in (Hardmeier, 2011), where a binary SVM classifier is trained to make predictions about the quality of the MT output. The datasets are syntactically analyzed using constituency and dependency parsers. The Subset Tree Kernel (Collins and Duffy, 2001) is used for the constituency trees, while the Partial Tree Kernel (Moschitti, 2006a) (Moschitti, 2006b) was judged as being more appropriate for the dependency trees. The evaluation shows that the combination between baseline features and the tree kernels achieves the best performance. These findings are further validated in Hardmeier et al. (2012) where a QE system is proposed based on a set of 82 explicit features combined with syntactic tree kernels.

Syntactic tree kernels for QE are also explored in Kaljahi et al. (2014), where a set of hand crafted constituency and dependency based features together with subset tree kernels applied on the constituency and dependency tree representations are used. The evaluation results demonstrate that the source constituency trees perform better than the target sentence constituency trees. This work is further extended in Kaljahi (2015), where multiple QE systems based on syntactic and semantic features are introduced.

The work presented in this paper differs from previous kernel approaches for QE by the innovative use of sequence kernels in addition to the previously utilized tree kernels. We extend on the previous kernel QE research by also making use of a back-translation of the target sentence in the computation of the kernels. While back-translations features have been previously utilized for QE (e.g.(Bechara et al., 2016)), their potential as an additional structural input representation for kernels has never been studied before. Furthermore, we exploit the potential of the scores of the kernel functions applied on the source and back-translation sentences as additional hard-coded features.

3 Methods and implementation

In this section details about the methodology and the implementation will be presented. First, tree and sequence kernels will be introduced, followed by the description of the implementation of these kernels in the context of QE. Finally, the machine learning platform used for implementing the QE systems will be presented.

3.1 Kernels for Quality Estimation

A kernel function computes the similarity between two structural representations without requiring the identification of the entire feature space (Moschitti, 2006a). To achieve this, the scalar product between vectors of substructure counts is computed in a vector space with a possibly infinite number of dimensions (Nguyen et al., 2009). Different kernel functions, depending on the type of structural input data they require, have been proposed including sequence, tree or graphs kernels. Tree kernels make use of tree representations for their computation, while sequence kernels calculate the similarity between the input sequence representations based on the number of common subsequences they share.

In the case of tree kernels, a series of algorithms have been proposed, e.g. in Collins and Duffy (2001) or Moschitti (2006a), based on the type of tree fragments (e.g. subsets, subtrees or partial trees) they take into consideration in their computation. On the other hand, sequence kernels have also been extensively studied in Bunescu and Mooney (2005) or Nguyen et al. (2009).

In this paper, we focus on the Partial Tree Kernel (Moschitti, 2006a) and the Subsequence Kernel (Bunescu and Mooney, 2005). The Partial Tree Kernel (PTK) was chosen because it is more flexible than the subtree or subset kernels in its calculation by taking partial subtrees into account. The Subsequence Kernel (SK) uses a dynamic pro-

	baseline features				baseline+new features			
	exact		not exact		exact		not exact	
System	**Pearson↑**	MAE ↓	**Pearson↑**	MAE ↓	**Pearson↑**	MAE ↓	**Pearson↑**	MAE ↓
SK src	0.408	0.145	0.416	0.143	0.413	0.144	0.422	0.143
SK src+mt	0.481	0.139	0.477	0.138	0.484	0.139	0.480	0.136
SK src+mt+mtbk	0.491	0.138	0.496	0.137	0.493	0.138	0.497	0.137
PTK src	0.449	0.137	0.452	0.138	0.459	0.137	0.463	0.137
PTK src+mt	0.495	0.133	0.499	0.133	0.50	0.133	0.505	0.132
PTK src+mt+mtbk	0.503	0.133	0.505	0.133	0.506	0.133	**0.509**	0.133
(PTK src+mt) + (SK src+mt)	0.488	0.137	0.487	0.136	0.490	0.137	0.488	0.136
(PTK src+mt+mtbk) + (SK src+mt+mtbk)	0.499	0.136	0.503	0.135	0.50	0.136	**0.504**	0.135
Baseline WMT	0.169	0.146						

Table 1: Evaluation results for the DE-EN dev set.

	baseline features				baseline+new features			
	exact		not exact		exact		not exact	
System	**Pearson↑**	MAE ↓	**Pearson↑**	MAE ↓	**Pearson↑**	MAE ↓	**Pearson↑**	MAE ↓
SK src	0.433	0.141	0.440	0.139	0.437	0.141	0.443	0.139
SK src+mt	0.478	0.138	0.483	0.137	0.480	0.138	0.484	0.139
SK src+mt+mtbk	0.466	0.142	0.478	0.140	0.467	0.142	0.479	0.140
PTK src	0.450	0.136	0.456	0.135	0.458	0.136	0.465	0.135
PTK src+mt	0.506	0.132	0.523	0.130	0.510	0.132	0.537	0.130
PTK src+mt+mtbk	0.491	0.137	0.501	0.137	0.493	0.137	0.503	0.137
(PTK src+mt) + (SK src+mt)	0.493	0.136	0.502	0.135	0.494	0.136	0.503	0.135
(PTK src+mt+mtbk) + (SK src+mt+mtbk)	0.478	0.141	0.488	0.140	0.479	0.141	0.489	0.140
Baseline WMT	0.260	0.140						

Table 2: Evaluation results for the DE-EN test set.

gramming approach to determine the number of common patterns between the two input sentences. In our experiments, the patterns taken into account were composed of the lexical items.

In order to use the tree kernel functions, the source and the target sentences were parsed using the Bohnet graph-based dependency parser (Bohnet, 2010), which was chosen because of its high accuracy. The data was first preprocessed by performing lemmatization and pos-tagging. Publicly available [1] pre-trained models were used for analyzing the source, target and back-translation sentences.

For learning using the Partial Tree Kernel, a transformation of the dependency parse tree is required, as introduced in Croce et al. (2011). We followed the lexical-centered-tree approach, where the grammatical relation and the pos-tag are encoded as the rightmost children of a dependency tree node. In the case of sequence kernels, the only preprocessing step applied was the tokenization of the input sentences. In order to investigate if prior lemmatization of the input sentences influences the results, we created two variants for each structural representation: an exact one containing the actual lexical items and a simplified non-exact one consisting of their corresponding lemmas.

Furthermore, we incorporated a back-translation of the target sentence as an additional structural input representation for both the tree kernels and the sequence kernels. The back-translation was obtained using the free online Google Machine Translation system [2]. We also exploited the full capability of the kernel functions by utilizing their explicit scores when applied on the source and back-translation sentences. We computed the scores for both the non-exact representations, and the exact ones. The scores were normalized using the formula from Croce et al. (2011)

$$score = \frac{K(T1, T2)}{\sqrt{K(T1, T1) \star K(T2, T2)}}$$

with *T1* and *T2* denoting the structural representations and *K* the type of kernel function applied.

3.2 KeLP (Kernel-based Learning Platform)

In our implementation, we applied the Partial Tree Kernel[3] and the Sequence Kernel [4] together with the epsilon-regression SVM implementations made available in the KeLP package (Filice et al., 2015b) (Filice et al., 2015a). KeLP (Kernel-based

[1] https://code.google.com/archive/p/mate-tools/downloads

[2] https://translate.google.com

[3] based on (Moschitti, 2006a)

[4] based on (Bunescu and Mooney, 2005)

	baseline features				baseline+new features			
	exact		not exact		exact		not exact	
System	**Pearson↑**	MAE ↓	**Pearson↑**	MAE ↓	**Pearson↑**	MAE ↓	**Pearson↑**	MAE ↓
SK src	0.446	0.137	0.434	0.140	0.450	0.137	0.436	0.140
SK src+mt	0.496	0.133	0.491	0.134	0.499	0.133	0.493	0.134
SK src+mt+mtbk	0.508	0.131	0.497	0.134	0.499	0.133	0.499	0.137
PTK src	0.467	0.134	0.469	0.134	0.476	0.134	0.477	0.133
PTK src+mt	0.516	0.130	0.524	0.129	0.480	0.134	0.530	0.129
PTK src+mt+mtbk	0.520	0.130	0.523	0.130	**0.524**	0.130	0.526	0.130
(PTK src+mt) + (SK src+mt)	0.508	0.131	0.516	0.132	0.510	0.131	0.518	0.132
(PTK src+mt+mtbk) + (SK src+mt+mtbk)	0.515	0.131	0.515	0.132	**0.516**	0.131	0.516	0.132
Baseline WMT	0.359	0.140						

Table 3: Evaluation results for the EN-DE dev set. The highlighted numbers correspond to the systems submitted to the shared task.

Learning Platform) is a Java Machine Learning library that provides the venue for implementing kernel based machine learning algorithms together with kernel functions. KeLP provides built-in support for multiple vectorial or structured data representations, which can be leveraged at the same time by combining different kernels into a single model. The package has a series of advantages, among them platform-independence, flexibility of use and its modularity that makes it easily extensible. The training of the QE prediction models was performed using the Support Vector Machine epsilon-Regression implementation with default parameters from the KeLP package. For the baseline systems a radial basis function (rbf) kernel was chosen, while for the other implemented QE systems the linear combination between the baseline features rbf kernel and the additional structural kernels was used.

4 Evaluation and results

4.1 Experimental setup

The evaluation was performed using the datasets released for the QE sentence-level shared task by the Second Conference On Machine Translation (WMT17) [5]. The data consists of tuples, containing the source segment, the target sentence and a manually post-edited version of the target sentence, together with their associated post-editing score.

The WMT17 dataset is composed of both English-German and German-English tuples. The English-German dataset, pertaining to the IT domain, consists of 23000 tuples for training, with additional 1000 instances for development. Two sets, comprised of 2000 units each, were made available for testing. On the other hand, the German-English dataset provides 25000 tuples for training, 1000 units for development and a test set consisting of 2000 instances, with the general domain categorized as Pharmaceutical. The QE baseline systems used for evaluation are based on the sets of 17 baseline features made available by the QE sentence-level shared task. They consist of surface features (e.g the number of tokens/punctuation marks in the source sentence), language model features (e.g LM probability of the source/target sentences), but also n-gram based features (e.g percentage of unigrams in quartile 4 of frequency (higher frequency words) in a corpus of the source language).

4.2 Results

The systems were evaluated based on their predicted scores using Pearson's correlation coefficient and the Mean Average Error (MAE), with the former being chosen as the primary method of evaluation for the WMT17 sentence-level QE task. We experimented with different model combinations and the results of the evaluation are presented in the tables that follow, where we have highlighted our submissions to the sentence level shared task. To better distinguish between models, the following QE system notation scheme was utilized: [*Kernel* [*level*]], where *Kernel* identifies the type of kernel used: PTK or SK and *level* represents the input type of sentence the kernel was applied to: source (marked with *src*), target (marked with *mt*) and back-translated target (marked with *mtbk*). The linear combination between the different kernel functions was marked with the plus sign. The systems can be categorized according to multiple criteria. The first one considers the presence of the new kernel features, which divides the systems into *baseline features* and *baseline+new*

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

	baseline features				baseline+new features			
	exact		not exact		exact		not exact	
System	**Pearson**↑	MAE ↓	**Pearson**↑	MAE ↓	**Pearson**↑	MAE ↓	**Pearson**↑	MAE ↓
SK src	0.448	0.131	0.443	0.132	0.456	0.131	0.451	0.132
SK src+mt	0.506	0.126	0.490	0.127	0.510	0.125	0.494	0.126
SK src+mt+mtbk	0.510	0.125	0.498	0.126	0.513	0.125	0.500	0.126
PTK src	0.461	0.129	0.439	0.130	0.474	0.128	0.452	0.129
PTK src+mt	0.508	0.124	0.500	0.124	0.515	0.124	0.508	0.123
PTK src+mt+mtbk	0.511	0.124	0.508	0.124	0.516	0.123	0.514	0.124
(PTK src+mt) + (SK src+mt)	0.517	0.125	0.508	0.125	0.520	0.126	0.511	0.125
(PTK src+mt+mtbk) + (SK src+mt+mtbk)	0.522	0.124	0.515	0.124	0.524	0.124	0.517	0.124
Baseline WMT	0.345	0.136						

Table 4: Evaluation results for the EN-DE 2016 test set

	baseline features				baseline+new features			
	exact		not exact		exact		not exact	
System	**Pearson**↑	MAE ↓	**Pearson**↑	MAE ↓	**Pearson**↑	MAE ↓	**Pearson**↑	MAE ↓
SK src	0.422	0.138	0.420	0.138	0.427	0.138	0.427	0.137
SK src+mt	0.482	0.132	0.470	0.133	0.485	0.132	0.473	0.133
SK src+mt+mtbk	0.494	0.131	0.482	0.132	0.495	0.131	0.483	0.132
PTK src	0.444	0.133	0.440	0.136	0.452	0.132	0.449	0.133
PTK src+mt	0.496	0.129	0.493	0.129	0.502	0.129	0.499	0.129
PTK src+mt+mtbk	0.504	0.129	0.505	0.129	0.508	0.129	0.509	0.128
(PTK src+mt) + (SK src+mt)	0.497	0.131	0.494	0.131	0.499	0.131	0.496	0.131
(PTK src+mt+mtbk) + (SK src+mt+mtbk)	0.508	0.130	0.506	0.130	0.509	0.130	0.508	0.130
Baseline WMT	0.387	0.135						

Table 5: Evaluation results for the EN-DE 2017 test set.

features systems. The second criterion is represented by the presence of the lemmatization in the pre-processing pipeline of the input sentences, which partitions the systems into *exact* and *not exact* ones.

A series of preliminary experiments was conducted which indicated that strictly structural kernel based methods could not capture all the relevant features for constructing a high performing QE system. Therefore, a combination between the baseline rbf kernel with additional structural kernels was implemented for the reported QE systems.

We can notice that all the systems, corresponding to both language pairs outperformed the baseline systems in terms of Pearson correlation. Of particular interest are the systems making use of the new kernel features, which succeeded in surpassing the corresponding systems that only used the baseline features.

The results also show that the addition of the back-translation as additional input data, proved on average beneficial for improving the correlation scores over systems that make use of only the source and target sentences as input data for the kernel functions.

In addition, we can observe that the sequence kernels based systems are highly performant in terms of Pearson's coefficient, albeit slightly worse on average than the tree kernels based implementations. This is a very important aspect, as the integration of sequence kernels into QE systems does not require additional external tools and therefore makes them well suited for low-resource language pairs, that might lack high-quality syntactic tools like parsers or taggers. Moreover, by employing a sequence kernel, the parsing of MT output is effectively bypassed. This constitutes an advantage as the parsing of target sentences often represents a challenging task due to the ungrammaticality of the MT generated output.

5 Conclusions and future work

In this paper, we presented our submission to the sentence level QE task, based on sequence and tree kernels. We have also investigated the performance of additional kernel-based features, as well as the benefit of incorporating a back-translation of the machine translation output as an additional input data representation, which to our knowledge has not been studied before. The results indicate that both ideas contribute useful additions to the baseline systems. We have also demonstrated that sequence kernels are a high performing method for predicting the quality of MT translations, that have the advantage of not requiring additional resources

for their computation.

We plan to further extend the current work by using constituency trees besides dependency trees for the computation of the tree kernels. We also plan to investigate if the choice of the MT system for the back-translation, affects the evaluation results. Lastly, more combination schemes between the tree and sequence kernels will be explored together with additional datasets and language pairs.

References

Eleftherios Avramidis. 2012. Comparative Quality Estimation: Automatic sentence-level ranking of multiple Machine Translation outputs. *Proceedings of COLING 2012: Technical Papers* pages 115–132.

Hanna Bechara, Carla Parra Escartin, Constantin Orasan, and Lucia Specia. 2016. Semantic Textual Similarity in Quality Estimation. *Baltic J. Modern Computing* pages 256–268.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. *The 23rd International Conference on Computational Linguistics (COLING 2010)* .

Razvan Bunescu and Raymond Mooney. 2005. Subsequence kernels for Relation Extraction. *Advances in Neural Information Processing Systems, Vol. 18: Proceedings of the 2005 Conference (NIPS)* .

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. *Proceedings of NIPS 2001* pages 625–632.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Structured lexical similarity via convolution kernels on dependency trees. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* pages 1034–1046.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for Relation Extraction. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics* .

Simone Filice, Giuseppe Castellucci, Roberto Basili, Giovanni Da San Martino, and Alessandro Moschitti. 2015a. KeLP: a Kernel-based Learning Platform in Java. *The workshop on Machine Learning Open Source Software (MLOSS): Open Ecosystems* .

Simone Filice, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2015b. KeLP: a kernel-based learning platform for Natural Language Processing. *Proceedings of ACL-IJCNLP 2015 System Demonstrations* pages 19–24.

Christian Hardmeier. 2011. Improving Machine Translation quality prediction with syntactic tree kernels. *Proceedings of the 15th Conference of the European Association for Machine Translation* pages 233–240.

Christian Hardmeier, Joakim Nivre, and Jorg Tiedemann. 2012. Tree kernels for Machine Translation Quality Estimation. *Proceedings of the 7th Workshop on Statistical Machine Translation* pages 109–113.

Rasoul Kaljahi. 2015. The role of syntax and semantics in Machine Translation and Quality Estimation of machine-translated user-generated content. *PhD Thesis* .

Rasoul Kaljahi, Jennifer Foster, Raphael Rubino, and Johann Roturier. 2014. Quality Estimation of English-French Machine Translation: A detailed study of the role of syntax. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* pages 2052–2063.

Anna Kozlova, Mariya Shmatova, and Anton Frolov. 2016. YSDA participation in the WMT16 Quality Estimation shared task. *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers* pages 793–799.

Alessandro Moschitti. 2006a. Efficient convolution kernels for dependency and constituent syntactic trees. *Proceedings of the 17th European Conference on Machine Learning* .

Alessandro Moschitti. 2006b. Making tree kernels practical for natural language learning. *Proceedings of the Eleventh International Conference of the European Association for Computational Linguistics* .

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2008. Tree kernels for Semantic Role Labeling. *Computational Linguistics* 34(2):193–224.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for Relation Extraction. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* pages 1378–1387.

Raphael Rubino, Jennifer Foster, Joachim Wagne, Johann Roturier, Rasul Samad Zadeh Kaljahi, and Fred Hollowood. 2012. DCU-Symantec submission for the WMT 2012 Quality Estimation task. *Proceedings of the Seventh Workshop on Statistical Machine Translation* pages 138–144.

Aliaksei Severyn, Massimo Nicosia, and Alessandro Moschitti. 2013. iKernels-Core: Tree kernel learning for Textual Similarity. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task* pages 53–58.

Rui Wang and Gnter Neumann. 2007. Recognizing Textual Entailment using a subsequence kernel method. *AAAI* 7:937–945.