FBK-UEdin participation to the WMT13 Quality Estimation shared-task

José G. C. de Souza
FBK-irst
University of Trento
Trento, Italy
desouza@fbk.eu

Christian Buck
School of Informatics
University of Edinburgh
Edinburgh, UK
christian.buck@ed.ac.uk

Marco Turchi, Matteo Negri
FBK-irst
Trento, Italy
{turchi,negri}@fbk.eu

Abstract

In this paper we present the approach and system setup of the joint participation of Fondazione Bruno Kessler and University of Edinburgh in the WMT 2013 Quality Estimation shared-task. Our submissions were focused on tasks whose aim was predicting sentence-level Human-mediated Translation Edit Rate and sentence-level post-editing time (Task 1.1 and 1.3, respectively). We designed features that are built on resources such as automatic word alignment, n-best candidate translation lists, back-translations and word posterior probabilities. Our models consistently overcome the baselines for both tasks and performed particularly well for Task 1.3, ranking first among seven participants.

1 Introduction

Quality Estimation (QE) for Machine Translation (MT) is the task of evaluating the quality of the output of an MT system without relying on reference translations. The WMT 2013 QE Shared Task defined four different tasks covering both word and sentence level QE. In this work we describe the Fondazione Bruno Kessler (FBK) and University of Edinburgh approach and system setup of our participation to the shared task. We developed models for two sentence-level tasks: Task 1.1: Scoring and ranking for post-editing effort, and Task 1.3: Predicting post-editing time.

The first task aims at predicting the Human-mediated Translation Edit Rate (HTER) (Snover et al., 2006) between a suggestion generated by a machine translation system and its manually post-edited version. The data set contains 2,754 English-Spanish sentence pairs post-edited by one translator (2,254 for training and 500 for test). We participated only in the scoring mode of this task.

The second task requires to predict the time, in seconds, that was required to post edit a translation given by a machine translation system. Participants are provided with 1,087 English-Spanish sentence pairs, source and suggestion, along with their respective post-edited sentence and post-editing time in seconds (803 data points for training and 284 for test).

For both tasks we applied supervised learning methods and made use of information about word alignments, n-best diversity scores, word posterior probabilities, pseudo-references, and back translation to train our models. In the remainder of this paper we describe the features designed for our participation (Section 2), the learning methods used to build our models (Section 3), the experiments that led to our submitted systems (Section 4), and we briefly conclude our experience in this evaluation task (Section 5).

2 Features

2.1 Word Alignment

Information about word alignments is used to extract quantitative (amount and distribution of the alignments) and qualitative features (importance of the aligned terms) under the assumption that features that explore what is aligned can bring improvements to tasks where sentence-level semantic relations need to be identified. Among the possible applications, Souza et al. (2013) recently investigated with success their application in Cross-lingual Textual Entailment for content synchronization (Mehdad et al., 2012; Negri et al., 2013).

For our experiments in both tasks we built word alignment models using the resources made available for the evaluation campaign. To train the word alignment models we used the MGIZA++ implementation (Gao and Vogel, 2008) of the IBM models (Brown et al., 1993) and the concatenation of Europarl, News Commentary, MultiUN, paral-
The word alignment features are divided into three main groups: AL, POS and IDF. The AL group regards quantitative information about aligned and unaligned words between source sentence (src) and machine translation output (tgt). The features of this group are computed for both src and tgt:

- proportion of aligned words;
- number of contiguous unaligned words normalized by the length of the sentence;
- length of the longest sequence of aligned/unaligned words normalized by the length of the sentence;
- average length of aligned/unaligned sequences of words;
- position of the first/last unaligned word normalized by the length of the sentence;
- proportion of aligned n-grams in the sentence.

To compute the features of the POS group we use part-of-speech (PoS) information for each word in src and tgt. Training and test data for both tasks were preprocessed with the TreeTagger (Schmid, 1995) and mapped to a more coarse-grained set of part-of-speech tags (P) based on the universal PoS tag set by Petrov et al. (2012). In this group there are two different types of features: one is computed for the alignments (the mapping between a word in src and a word in tgt) and the other is computed for aligned words (words in src that are aligned to one or more words in tgt and vice-versa). The features computed over the alignments are:

- proportion of alignments connecting words with the same PoS tag;
- proportion of alignments connecting words with the same PoS tag for each tag p ∈ P.

The features implemented for aligned words are:

- proportion of aligned words tagged with p in the sentence (p ∈ P). This feature is processed for both src and tgt;
- proportion of words in src aligned with words in tgt that share the same PoS tag (and vice-versa);
- proportion of words tagged with p in src and that are aligned to words with the same tag p in tgt (and vice-versa). This is done for every p ∈ P.

The last group, IDF, is composed by one feature that explores the notion of inverse document frequency as another source of qualitative information. The idea is that rare words (with higher IDF) are more informative than frequent words. The IDF scores for each word are calculated for English and Spanish on each side of the parallel corpora used to build the alignment models. This feature is calculated for both src and tgt (at test stage, the average IDF value of each language is assigned to unseen terms):

- summation of the IDF scores of aligned words in src divided by the sum of IDF scores of the aligned words in tgt (and vice-versa).

Preliminary experiments have been executed to find the best word alignment algorithm for each task. We explored three different word alignment algorithms: the hidden Markov model (HMM) (Vogel et al., 1996) and IBM models 3 and 4 (Brown et al., 1993). We also tried three symmetrization models (Koehn et al., 2005): union, intersection, and grow-diag-final-and, a more complex symmetrization method which combines intersection with some alignments from the union. The best alignment and symmetrization combination found for Task 1.1 was IBM4 with intersection and for task 1.3 was HMM with intersection. These experiments were carried out in 10-fold cross-validation on the training set and used only the alignment features.

2.2 N-best Diversity scores

Our n-best diversity features are based on the intuition that a large number of possible choices generally leads to more errors. While a similar notion can be expressed locally by counting the translation options for each word or phrase, we consider n-best lists as a good approximation of the search space. This allows us to circumvent problems associated with the local measures, such as ambiguous alignment and segmentation, and limitations
of using the search graph directly such as the inability compute edit distance between hypotheses. Thus, to quantify the coherence of translation options we compute a (symmetrical) matrix of pairwise Levenshtein distances, either on token or character level, for n-best lists of size up to 100k using the baseline system and the systems we describe in Section 2.4. For this matrix the following features are produced:

1. The index of the central hypothesis, i.e. the translation with the minimum average distance to all other entries.
2. The average edit distance between the central hypothesis and all other entries normalized by the length of top scoring hypothesis.
3. Edit distance between top scoring and central hypothesis
4. Number of hypotheses with an edit distance to the top-scoring hypothesis below a set threshold.

2.3 Word Posterior Probabilities

Following previous work on word posterior probabilities (WPPs) (Ueffing et al., 2003) we computed the sequence of edit operations needed to transform the MT suggestion into all entries of an n-best list in which we normalized the logarithmic model scores to resemble probabilities. Tokens are considered incorrect is the operation is either insert or substitute, otherwise the probability of the hypothesis counts towards the correctness of the word. These word-level features were then normalized by taking the geometric mean of the individual probabilities. We did this for all systems described in Section 2.4 and varying sizes of n between 10 and 100k.

2.4 Pseudo-references and back-translation

Motivated by the success of pseudo-reference features (Soricut et al., 2012) we employed three additional MT systems: one similar to the original system but trained on more data, a hierarchical phrase-based system, and a Spanish-English system to translate back into English. All models have been estimated using publicly available software (SRILM (Stolcke, 2002), Moses (Koehn et al., 2007)), and corpora (Europarl, News Commentary, MultiUN, Gigaword). Using the predictions of the English-Spanish systems as pseudo-references and likewise the original source as reference for the back-translation system we computed a number of automatic metrics including BLEU (Papineni et al., 2002), GTM (Turian et al., 2003), PER (Tillmann et al., 1997), TER (Snover et al., 2006) and Meteor (Denkowski and Lavie, 2011).

3 Learning algorithms

To build our models using the features presented in Section 2 we tried different learning algorithms. After some preliminary experiments for both tasks we decided to use mainly two: support vector machines (SVM) and extremely randomized trees (Geurts et al., 2006). For all experiments presented in this paper we use the Scikit-learn (Pedregosa et al., 2011) implementations of the above algorithms.

In preliminary experiments we noticed that the number of features that we were using for both tasks was leading to poor results when using the SVM regression (SVR) models. In order to cope with this problem we performed feature selection prior to the SVM regression training. For that we used Randomized Lasso, or stability selection (Meinshausen and Bühlmann, 2010). It resamples the training data several times and fits a Lasso regression model on each sample. Features that appear in a given number of samples are retained. Both the fraction of the data to be sampled and the threshold to select the features can be configured. In our experiments we set the sampling fraction to 75%, the selection threshold to 25% and the number of re-samples to 200.

To optimize the SVR with radial basis function (RBF) kernel hyper-parameters we used random search (Bergstra and Bengio, 2012) instead of the traditional grid search procedure. We found random search to be as efficient or better than grid search and it drastically reduced the time required to compute the best parameter combination.

Finally, we trained an extremely randomized forest, i.e. an ensemble of extremely randomized trees. Each tree can be parameterized differently. The results of the individual trees are combined by averaging their predictions. When a tree is built,
Table 1: Experiments results for Task 1.1 on 10-fold cross-validation. “Base” are the 17 baseline features. “All” corresponds to all the features described in Section 2 in a total of 141 features. “SVR” is support vector regression, “RL” is randomized Lasso and “ET” is extremely randomized trees. MAE stands for the average mean absolute error and RMSE is the root mean squared error. Parameters for SVR are $C$, $\epsilon$, $\gamma$ and for ET is the number of estimators.

<table>
<thead>
<tr>
<th>System</th>
<th>Features</th>
<th>MAE</th>
<th>RMSE</th>
<th>Predict. Interval</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVR</td>
<td>Base</td>
<td>0.127</td>
<td>0.163</td>
<td>[0.046, 0.671]</td>
<td>347.5918, 0.001, 0.0001</td>
</tr>
<tr>
<td>SVR</td>
<td>Base + All</td>
<td>0.121</td>
<td>0.155</td>
<td>[0.090, 0.714]</td>
<td>0.4052, 0.0753, 0.0010</td>
</tr>
<tr>
<td>RL + SVR</td>
<td>Sel(Base + All)</td>
<td>0.119</td>
<td>0.1534</td>
<td>[0.084, 0.745]</td>
<td>40.5873, 0.0484, 0.0002</td>
</tr>
<tr>
<td>ET</td>
<td>Base + All</td>
<td>0.123</td>
<td>0.156</td>
<td>[0.142, 0.708]</td>
<td>100</td>
</tr>
<tr>
<td>ET</td>
<td>Base + All</td>
<td>0.122</td>
<td>0.155</td>
<td>[0.164, 0.712]</td>
<td>1000</td>
</tr>
</tbody>
</table>

During our experiments with the training set, the best model for Task 1.1 was the combination of randomized Lasso feature selection with SVR (0.119 MAE). The extremely randomized trees presented results around 0.12 MAE worse than the figures obtained by the SVR models. Results obtained for Task 1.1 are summarized in Table 1.

As for Task 1.3, training results are presented in Table 2. The best model combines feature selection (randomized Lasso) with SVR. During training it obtained the lowest average MAE (38.6). Compared to the models built with extremely randomized trees, the prediction interval of this system is narrower. This indicates that the tree-based models cover a wider range of data points than the SVR-based models.

In the official results released by the organizers our submissions had close performances for Task 1.1. The difference between the SVR and the extremely randomized tree models is very small (around 0.0012 MAE points). For Task 1.3 our best submission is the one based on ensembles of trees, a trend that was not observed during training. Our hypothesis is that the tree-based ensemble model was capable of generalizing the training data better than the SVR-based ones and that despite the low number of employed features the latter was prone to overfitting.

Table 3 presents the official evaluation numbers for both tasks.

4.1 Feature analysis

To gain some insight about the relevance of the features we explored in our submissions, we compared the output of the randomized Lasso with the most important features computed by the extremely randomized tree algorithm. Below we present the features that appear in the intersection

---

Given by $MAE = \frac{1}{N} \sum_{i=1}^{N} |H(s_i) - V(s_i)|$, where $H(s_i)$ is the hypothesis score for the entry $s_i$ and $V(s_i)$ is the gold standard value for $s_i$ in a dataset with $N$ entries.
Table 2: Experiments results for Task 1.3 on 10-fold cross-validation. “Base” are the 17 baseline features. “All” corresponds to all the features described in Section 2 in a total of 141 features. “SVR” is support vector regression, “RL” is randomized Lasso and “ET” is extremely randomized trees. MAE stands for the average mean absolute error and RMSE is the root mean squared error. Parameters for SVR are C, ϵ, γ and for ET is the number of estimators.

<table>
<thead>
<tr>
<th>System</th>
<th>Features</th>
<th>MAE</th>
<th>RMSE</th>
<th>Predict. Interval</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVR</td>
<td>Base</td>
<td>41.3</td>
<td>69.2</td>
<td>[5.6, 315.7]</td>
<td>138.7359, 2.3331, 0.0185</td>
</tr>
<tr>
<td>SVR</td>
<td>Base + All</td>
<td>40.2</td>
<td>70.6</td>
<td>[8.6, 335.6]</td>
<td>308.3817, 0.2194, 0.0009</td>
</tr>
<tr>
<td>RL + SVR</td>
<td>Sel(Base + All)</td>
<td>38.6</td>
<td>69.1</td>
<td>[11.5, 332.0]</td>
<td>161.5705, 7.3370, 0.0460</td>
</tr>
<tr>
<td>ET</td>
<td>Base + All</td>
<td>44.1</td>
<td>72.2</td>
<td>[11.9, 446.2]</td>
<td>100</td>
</tr>
<tr>
<td>ET</td>
<td>Base + All</td>
<td>43.7</td>
<td>72.0</td>
<td>[12.6, 446.2]</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table 3: Official results for tasks 1.1 and 1.3 on the test set.

<table>
<thead>
<tr>
<th>System</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Official Baseline</td>
<td>0.1491</td>
<td>0.1822</td>
</tr>
<tr>
<td>RL + SVR</td>
<td>0.1450</td>
<td>0.1773</td>
</tr>
<tr>
<td>ET</td>
<td>0.1438</td>
<td>0.1768</td>
</tr>
<tr>
<td>Task 1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Official Baseline</td>
<td>51.93</td>
<td>93.35</td>
</tr>
<tr>
<td>RL + SVR</td>
<td>47.92</td>
<td>86.66</td>
</tr>
<tr>
<td>ET</td>
<td>47.52</td>
<td>82.60</td>
</tr>
</tbody>
</table>

For Task 1.3, the randomized Lasso selection reduced the input feature vector from 141 features to 19. We compared these features with the 19 most important features computed by the extremely randomized tree algorithm. As above the intersection of both sets utilizes many resources:

- proportion of the last unaligned word normalized by the length of the sentence for src and tgt;
- summation of the IDF scores of aligned words in tgt divided by the summation of IDF scores of the aligned words in src;
- length of the longest sequence of unaligned words normalized by the length of the src;
- percentage of bigrams in the 4th quartile of frequency of the source language corpus;
- percentage of trigrams in the 4th quartile of frequency of the source language corpus;
- proportion of alignments connecting words with the same PoS tag;
- proportion of aligned words in src.

For Task 1.3, the feature selection algorithm retained 29 out of 112 features. We take the intersection of this set with the 29 most relevant features computed by the ensemble tree-based method. This selection comes from features based on different resources:

- proportion of words in src aligned with words in tgt that share the same PoS tag;
- average number of translations per source word according to IBM Model 1 thresholded $P(t|s) > 0.01$;
- average number of translations per source word according to IBM Model 1 thresholded $P(t|s) > 0.2$;
- average source sentence token length;
- number of times the top-scoring hypothesis is repeated in an 10k-best list;
- position of the first unaligned word normalized by the length of the sentence for src and tgt;
• number of occurrences of the target word within the target hypothesis averaged for all words in the hypothesis;
• percentage of bigrams in the 4th quartile of frequency of the source language corpus;
• percentage of trigrams in the 4th quartile of frequency of the source language corpus;
• number of contiguous unaligned words in tgt normalized by the length of tgt.

5 Conclusion
This paper presented the participation of FBK and University of Edinburgh to the WMT 2013 Quality Estimation shared task. Our approach explored features based on word alignment, n-best diversity scores, pseudo-references and back-translations, and word posterior probabilities. We experimented with two different learning methods, SVR and extremely randomized trees for predicting sentence-level post-editing time and HTER.

Our submitted systems were particularly successful for predicting sentence-level post-editing time, ranking 1st among seven participants. The submitted models for predicting HTER consistently overcame the baseline for the task. In addition to the description of our approach and system setup, we presented a first analysis of the features used in our models with the objective of assessing the importance of the features used either for predicting time or HTER.

6 Acknowledgments
This work was partially funded by the European Commission under the project MateCat, Grant 287688. The authors want to thank Philipp Koehn for training two of the models used in Section 2.2.

References


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


