Machine learning methods for comparative and time-oriented Quality Estimation of Machine Translation output

Eleftherios Avramidis and Maja Popović
German Research Center for Artificial Intelligence (DFKI GmbH)
Language Technology Lab
Alt Moabit 91c, 10559 Berlin
eleftherios.avramidis@dfki.de and maja.popovic@dfki.de

Abstract
This paper describes a set of experiments on two sub-tasks of Quality Estimation of Machine Translation (MT) output. Sentence-level ranking of alternative MT outputs is done with pairwise classifiers using Logistic Regression with black-box features originating from PCFG Parsing, language models and various counts. Post-editing time prediction uses regression models, additionally fed with new elaborate features from the Statistical MT decoding process. These seem to be better indicators of post-editing time than black-box features. Prior to training the models, feature scoring with ReliefF and Information Gain is used to choose feature sets of decent size and avoid computational complexity.

1 Introduction
During the recent years, Machine Translation (MT) has reached levels of performance which allow for its integration into real-world translation workflows. Despite the high speed and various advantages of this technology, the fact that the MT results are rarely perfect and often require manual corrections has raised a need to assess their quality, predict the required post-editing effort and compare outputs from various systems on application time. This has been the aim of current research on Quality Estimation, which investigates solutions for several variations of such problems.

We describe possible solutions for two problems of MT Quality Estimation, as part of the 8th Shared Task on Machine Translation: (a) sentence-level quality ranking (1.2) of multiple translations of the same source sentence and (b) prediction of post-editing time (1.3). We present our approach on acquiring (section 2.1) and selecting features (section 2.2), we explain the generation of the statistical estimation systems (section 2.3) and we evaluate the developed solutions with some of the standard metrics (section 3).

2 Methods: Quality Estimation as machine learning
These two Quality Estimation solutions have been seen as typical supervised machine learning problems. MT output has been given to humans, so that they perform either (a) ranking of the multiple MT system outputs in terms of meaning or (b) post-editing of single MT system output, where time needed per sentence is measured. The output of these tasks has been provided by the shared task organizers as a training material, whereas a small keep-out set has been reserved for testing purposes.

Our task is therefore to perform automatic quality analysis of the translation output and the translation process in order to provide features for the supervised machine learning mechanism, which is then trained over the corresponding to the respective human behaviour. The task is first optimized in a development phase in order to produce the two best shared task submissions for each task. These are finally tested on the keep-out set so that their performance is compared with the ones submitted by all other shared-task participants.

2.1 Feature acquisition
We acquire two types of sentence-level features, that are expected to provide hints about the quality of the generated translation, depending on whether they have access to internal details of the MT decoding process (glass-box) or they are only derived from characteristics of the processed and generated sentence text (black-box).
2.1.1 Black-box features
Features of this type are generated as a result of automatic analysis of both the source sentence and the MT output (when applicable), whereas many of them are already part of the baseline infrastructure. For all features we also calculate the ratios of the source to the target sentence. These features include:

PCFG Features: We parse the text with a PCFG grammar (Petrov et al., 2006) and we derive the counts of all node labels (e.g. count of VPs, NPs etc.), the parse log-likelihood and the number of the n-best parse trees generated (Avramidis et al., 2011).

Rule-based language correction is a result of hand-written controlled language rules, that indicate mistakes on several pre-defined error categories (Naber, 2003). We include the number of errors of each category as a feature.

Language model scores include the smoothed n-gram probability and the n-gram perplexity of the sentence.

Count-based features include count and percentage of tokens, unknown words, punctuation marks, numbers, tokens which do or do not contain characters “a-z”; the absolute difference between number of tokens in source and target normalized by source length, number of occurrences of the target word within the target hypothesis averaged for all words in the hypothesis (type/token ratio).

Source frequency: A set of eight features includes the percentage of uni-grams, bi-grams and tri-grams of the processed sentence in frequency quartiles 1 (lower frequency words) and 4 (higher frequency words) in the source side of a parallel corpus (Callison-Burch et al., 2012).

Contrastive evaluation scores: For the ranking task, each translation is scored with an automatic metric (Papineni et al., 2002; Lavie and Agarwal, 2007), using the other translations as references (Soricut et al., 2012).

2.1.2 Glass-box features
Glass-box features are available only for the time-prediction task, as a result of analyzing the verbose output of the Minimum Bayes Risk decoding process.

Counts from the best hypothesis: Count of phrases, tokens, average/minimum/maximum phrase length, position of longest and shortest phrase in the source sentence; count of words unknown to the phrase table, average number of unknown words first/last position of an unknown word in the sentence normalized to the number of tokens, variance and deviation of the position of the unknown words.

Log probability (pC) and future cost estimate (c) of the phrases chosen as part of the best translation: minimum and maximum values and their position in the sentence averaged to the number of sentences, and also their average, variance, standard deviation; count of the phrases whose probability or future cost estimate is lower and higher than their standard deviation; the ratio of these phrases to the total number of phrases.

Alternative translations from the search path of the decoder: average phrase length, average of the average/variance/standard deviation of phrase log probability and future cost estimate, count of alternative phrases whose log probability or future cost estimate is lower and higher than their standard deviation.

2.2 Feature selection
Feature acquisition results in a huge number of features. Although the machine learning mechanisms already include feature selection or regularization, huge feature sets may be unusable for training, due to the high processing needs and the sparsity or noise they may infer. For this purpose we first reduce the number of features by scoring them with two popular correlation measurement methods.

2.2.1 Information gain
Information gain (Hunt et al., 1966) estimates the difference between the prior entropy of the classes and the posterior entropy given the attribute values. It is useful for estimating the quality of each attribute but it works under the assumption that features are independent, so it is not suitable when strong feature inter-correlation exists. Information gain is only used for the sentence ranking task after discretization of the feature values.

2.2.2 ReliefF
ReliefF assesses the ability of each feature to distinguish between very similar instances from dif-
frent classes (Kononenko, 1994). It picks up a number of instances in random and calculates a feature contribution based on the nearest hits and misses. It is a robust method which can deal with incomplete and noisy data (Robnik-Šikonja and Kononenko, 2003).

2.3 Machine learning algorithms

Machine learning is performed for the two sub-tasks using common pairwise classification and regression methods, respectively.

2.3.1 Ranking with pairwise binary classifiers

For the sub-task on sentence-ranking we used pairwise classification, so that we can take advantage of several powerful binary classification methods (Avramidis, 2012). We used logistic regression, which optimizes a logistic function to predict values in the range between zero and one (Cameron, 1998), given a feature set \( X \):

\[
P(X) = \frac{1}{1 + e^{-1(a+bX)}}
\]  

(1)

The logistic function is fitted using the Newton-Raphson algorithm to iteratively minimize the least squares error computed from training data (Miller, 2002). Experiments are repeated with two variations of Logistic Regression concerning internal features treatment: Stepwise Feature Set Selection (Hosmer, 1989) and L2-Regularization (Lin et al., 2007).

2.3.2 Regression

For the sub-task on post-editing time prediction, we experimented with several regression methods, such as Linear Regression, Partial Least Squares (Stone and Brooks, 1990), Multivariate Adaptive Regression Splines (Friedman, 1991), LASSO (Tibshirani, 1996), Support Vector Regression (Basak et al., 2007) and Tree-based regressors. Indicatively, Linear regression optimizes co-efficient \( \beta \) for predicting a value \( y \), given a feature vector \( X \):

\[
y = X\beta + \varepsilon
\]  

(2)

2.4 Evaluation

The ranking task is evaluated by measuring correlation between the predicted and the human ranking, with the use of Kendall tau (Kendall, 1938) including penalization of ties. We additionally consider two more metrics specialized in ranking tasks: Mean Reciprocal Rank - MRR (Voorhees, 1999) and Normalized Discounted Cumulative Gain - NDGC (Järvelin and Kekäläinen, 2002), which give better scores to models when higher ranks (i.e. better translations) are ordered correctly, as these are more important than lower ranks.

The regression task is evaluated in terms of Root Mean Square Error (RMSE) and Mean Average Error (MAE).

3 Experiment and Results

3.1 Implementation

Relieff is implemented for \( k=5 \) nearest neighbours sampling \( m=100 \) reference instances. Information gain is calculated after discretizing features into \( n=100 \) values.

N-gram features are computed with the SRILM toolkit (Stolcke, 2002) with an order of 5, based on monolingual training material from Europarl (Koehn, 2005) and News Commentary (Callison-Burch et al., 2011). PCFG parsing features are generated on the output of the Berkeley Parser (Petrov and Klein, 2007) trained over an English, a German and a Spanish treebank (Taulé et al., 2008). The open source language tool\(^1\) is used to annotate source and target sentences with language suggestions. The annotation process is organised with the Ruffus library (Goodstadt, 2010) and the learning algorithms are executed using the Orange toolkit (Demšar et al., 2004).

3.2 Sentence-ranking

The sentence-ranking sub-task has provided training data for two language pairs, German-English and English-Spanish. For both sentence pairs, we train the systems using the provided annotated data sets WMT2010, WMT2011 and WMT2012, while the data set WMT2009 is used for the evaluation during the development phase. Data sets are analyzed with black-box feature generation. For each language pair, the two systems with the highest correlation are submitted.

We start the development with two feature sets that have shown to perform well in previous experiments: \#24 (Avramidis, 2012) including features from PCFG parsing, and \#31 which is the baseline feature set of the previous year’s shared task (Callison-Burch et al., 2012) and we combine them (\#33). Additionally, we create feature sets by

\(^1\)Open source at http://languagetool.org
scoring features with ReliefF (features #41x) and Information Gain (#42). Many combinations of all the above feature-sets are tested and the most important of them are shown in Table 1. Feature sets are described briefly in Table 2.

For **German-English**, we experiment with 14 feature sets, using both variations of Logistic Regression. The two highest tau scores are given by Stepwise Feature Set Selection using feature sets #33 and #24. We see that although baseline features #31 alone have very low correlation, when combined with previously successful #24, provide the best system in terms of tau. Feature set #431 (which combines the 15 features scored higher with ReliefF, the 15 features scored higher with Information Gain and the feature set #24) succeeds pretty well on the additional metrics MRR and NDGC, but it provides slightly lower tau correlation.

For **English-Spanish**, the correlation of the produced systems is significantly lower and it appears that the L2-regularized logistic regression performs better as classification method. We experiment with 24 feature sets, after more scoring with ReliefF and Inf. Gain. Surprisingly enough, Kendall tau correlation indicates that the best model is trained only with features based on counts of numbers and punctuation, combined with **contrastive BLEU score**. This seems to rather overfit a peculiarity of the particular development set and indeed performs much lower on the final test set of the shared task (tau=0.04). The second best feature set (#431) has been described above and luckily generalizes better on an unknown set. It is interesting to see that this issue would have been avoided, if the decision was taken based on the ranking metrics MRR and NDGC, which prioritize other feature sets. We assume that further work is needed to see whether these measures are more expressive and reliable than Kendall tau for similar tasks.

The fitted $\beta$ coefficients (in tables 3 and 4) give an indication of the importance of each feature (see equation 1), for each language pair. In both language pairs, target-side features prevail upon other features. On the comparison of the models for the two language pairs (and the $\beta$ coefficients as well) we can see that the model settings and performance may vary from one language pair to another. This also requires further investigation, given that Kendall tau and the other two metrics indicate different models as the best ones.

The fact that the German-English set is better fitted with Stepwise Feature Set Selec-
set features

#24 From previous work (Avramidis, 2012):
[s+t]: PCFG\log, count of: unknown words, tokens, PCFG trees, VPs

[t]: pseudoMETEOR

#31 Baseline from WMT12 (Callison-Burch et al., 2012)
[s+t]: tokens\avg, \lmprob, count of: commas, dots, tokens, avg translations per source word

[s]: avg freq. of low and high freq. bi-grams/tri-grams, % of distinct uni-grams in the corpus

[t]: type/token radio

#32 All 50 “vanilla” features provided by shared-task baseline software “Quest”

#411 ReliefF best 5 features

[s+t]: % of numbers, difference between periods of source and target (plain and averaged)

[t]: pseudoBLEU

Table 2: Description of most important feature sets for task 1.2, before internal feature selection of Logistic Regression is applied. [s] indicates source, [t] indicates target

<table>
<thead>
<tr>
<th>set</th>
<th>de-en</th>
<th>en-es</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StepFSS</td>
<td>L2reg</td>
</tr>
<tr>
<td>#24</td>
<td>0.28</td>
<td>0.25</td>
</tr>
<tr>
<td>#33</td>
<td>0.29</td>
<td>0.26</td>
</tr>
<tr>
<td>#411</td>
<td>0.22</td>
<td>0.17</td>
</tr>
<tr>
<td>#431</td>
<td>0.27</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 5: Higher Kendall tau correlation (on the dev. set) is achieved on German-English by using Stepwise Feature Set Selection, whereas on English-Spanish by using L2-regularization

Figure 1: Graphical representation of the values predicted by the linear regression model with feature set #6 (blue) against the actual values of the development set (red)

3.3 Post-editing time prediction

The training for the model predicting post-editing time is performed over the entire given data set and the evaluation is done with 10-fold cross-validation. We evaluated 8 feature sets with 6 regression methods each, ending up with 48 experiments.

The evaluation of the most indicative regression models (two best performing ones per feature set) can be seen in Table 6. We start with a glass-box feature set, scored with ReliefF and consequently add black-box features. We note the models that have the lowest Root Mean Square Error and Mean Average Error.

Our best model seems to be the one built linear regression using feature set #6. This feature set is chosen by collecting the 17 best features as scored by ReliefF and includes both black-box and glass-box features. How well this model fits the development test is represented in Figure 1.

The second best feature set (#8) includes 29 glass-box features with the highest absolute ReliefF, joined with the black-box features of the successful feature set #6.

More details about the contribution of the most important features in the linear regression (equation 2) can be seen in table 7, where the fitted coefficients of each feature are given. The vast majority of the best contributing features are glass-box features. Some draft conclusions out of the coefficients may be that post-editing time is lower when:

333
• the longest of the source phrases used for producing the best hypothesis appears closer to the end of the sentence

• the phrases with the highest and the lowest probability appear closer to the end of the translated sentence

• there are more determiners in the source and/or less determiners in the translation

• there are more verbs in the translation and/or less verbs in the source

• there are fewer alternative phrases with very high probability

Further conclusions can be drawn after examining these observations along with the exact operation of the statistical MT system, which is subject to further work.

4 Conclusion

We describe two approaches for two respective problems of quality estimation, namely sentence-level ranking of alternative translations and prediction of time for post-editing MT output. We present efforts on compiling several feature sets and we examine the final contribution of the features after training Machine Learning models. Elaborate decoding features seem to be quite helpful for predicting post-editing time.

<table>
<thead>
<tr>
<th>id</th>
<th>feature set</th>
<th>method</th>
<th>RMSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>20 glass-box features with highest absolute ReliefF</td>
<td>MARS</td>
<td>91.54</td>
<td>59.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SVR</td>
<td>93.57</td>
<td>55.87</td>
</tr>
<tr>
<td>#2</td>
<td>9 glass-box features with highest positive ReliefF</td>
<td>Lasso</td>
<td>83.20</td>
<td>51.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear</td>
<td>83.32</td>
<td>51.72</td>
</tr>
<tr>
<td>#3</td>
<td>16 glass-box features with highest positive ReliefF</td>
<td>Lasso</td>
<td>77.54</td>
<td>47.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear</td>
<td>77.60</td>
<td>47.27</td>
</tr>
<tr>
<td>#4</td>
<td>22 glass-box features with highest positive ReliefF</td>
<td>Lasso</td>
<td>76.05</td>
<td>46.37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear</td>
<td>76.17</td>
<td>46.48</td>
</tr>
<tr>
<td>#5</td>
<td>Combination of feature sets #1 and #2</td>
<td>MARS</td>
<td>91.54</td>
<td>59.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SVR</td>
<td>93.57</td>
<td>55.87</td>
</tr>
<tr>
<td>#6</td>
<td>17 features of any type with highest positive ReliefF</td>
<td>Linear</td>
<td>74.70</td>
<td>45.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lasso</td>
<td>74.75</td>
<td>44.99</td>
</tr>
<tr>
<td>#8</td>
<td>Combination of #5 and #6 + counts of tokens</td>
<td>Lasso</td>
<td>75.14</td>
<td>44.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLS</td>
<td>77.63</td>
<td>47.48</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>id</th>
<th>feature set</th>
<th>method</th>
<th>RMSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>#6</td>
<td>First submission</td>
<td>Linear</td>
<td>84.27</td>
<td>52.41</td>
</tr>
<tr>
<td>#8</td>
<td>Second submission</td>
<td>PLS</td>
<td>88.34</td>
<td>53.49</td>
</tr>
<tr>
<td></td>
<td>Best models</td>
<td></td>
<td>82.60</td>
<td>47.52</td>
</tr>
</tbody>
</table>

Table 6: Development and submitted experiments for task 1.3

<table>
<thead>
<tr>
<th>feature</th>
<th>( \beta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>best hyp: position of the longest aligned phrase in the source sentence averaged to the number of phrases</td>
<td>-16.652</td>
</tr>
<tr>
<td>best hyp: position of phrase with highest prob. averaged to the num. of phrases</td>
<td>-14.824</td>
</tr>
<tr>
<td>source: number of determiners</td>
<td>-9.312</td>
</tr>
<tr>
<td>best hyp: number of determiners</td>
<td>6.189</td>
</tr>
<tr>
<td>best hyp: position of phrase with lowest prob. averaged to the num. of phrases</td>
<td>-5.261</td>
</tr>
<tr>
<td>best hyp: position of phrase with lowest future cost estimate averaged to the number of phrases</td>
<td>-4.282</td>
</tr>
<tr>
<td>best hyp: number of verbs</td>
<td>-2.818</td>
</tr>
<tr>
<td>best hyp: position of phrase with highest future cost estimate averaged to the number of phrases</td>
<td>1.002</td>
</tr>
<tr>
<td>search: number of alternative phrases with very low future cost est.</td>
<td>-0.528</td>
</tr>
<tr>
<td>source: number of verbs</td>
<td>0.467</td>
</tr>
<tr>
<td>search: number of alternative phrases with very high probability</td>
<td>0.355</td>
</tr>
<tr>
<td>search: total num. of translation options</td>
<td>-0.153</td>
</tr>
<tr>
<td>search: number of alternative phrases with very high future cost estimate</td>
<td>-0.142</td>
</tr>
<tr>
<td>best hyp: number of parse trees</td>
<td>0.007</td>
</tr>
<tr>
<td>source: number of parse trees</td>
<td>0.002</td>
</tr>
<tr>
<td>search: total number of hypotheses</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Table 7: Linear regression coefficients for feature set #6 indicate the contribution of each feature in the fitted model
Acknowledgments

This work has been developed within the TaraXU project, financed by TSB Technologiestiftung Berlin – Zukunftsfonds Berlin, co-financed by the European Union – European fund for regional development. Many thanks to Prof. Hans Uszkoreit for the supervision, Dr. Aljoscha Burchardt, and Dr. David Vilar for their useful feedback and to Lukas Poustka for his technical help on feature acquisition.

References


