Evaluate with Confidence Estimation: Machine ranking of translation outputs using grammatical features

Eleftherios Avramidis, Maja Popovic, David Vilar, Aljoscha Burchardt
German Research Center for Artificial Intelligence (DFKI)
Language Technology (LT), Berlin, Germany
name.surname@dfki.de

Abstract

We present a pilot study on an evaluation method which is able to rank translation outputs with no reference translation, given only their source sentence. The system employs a statistical classifier trained upon existing human rankings, using several features derived from analysis of both the source and the target sentences. Development experiments on one language pair showed that the method has considerably good correlation with human ranking when using features obtained from a PCFG parser.

1 Introduction

Automatic evaluation metrics for Machine Translation (MT) have mainly relied on analyzing both the MT output against (one or more) reference translations. Though, several paradigms in Machine Translation Research pose the need to estimate the quality through many translation outputs, when no reference translation is given (*n*-best rescoring of SMT systems, system combination etc.). Such metrics have been known as *Confidence Estimation metrics* and quite a few projects have suggested solutions on this direction. With our submission to the Shared Task, we allow such a metric to be systematically compared with the state-of-the-art reference-aware MT metrics.

Our approach suggests building a Confidence Estimation metric using already existing human judgments. This has been motivated by the existence of human-annotated data containing comparisons of the outputs of several systems, as a result of the evaluation tasks run by the Workshops on Statistical Machine Translation (WMT) (**?**; **?**; **?**). This amount of data, which has been freely available for further research, gives an opportunity for applying machine learning techniques to model the human annotators' choices. Machine Learning methods over previously released evaluation data have been already used for tuning complex statistical evaluation metrics (e.g. SVM-Rank in **?**)). Our proposition is similar, but works without reference translations. We develop a solution of applying machine learning in order to build a statistical classifier that performs similar to the human ranking: it is trained to rank several MT outputs, given analysis of possible qualitative criteria on both the source and the target side of every given sentence. As qualitative criteria, we use statistical features indicating the quality and the grammaticality of the output.

2 Automatic ranking method

2.1 From Confidence Estimation to ranking

Confidence estimation has been seen from the Natural Language Processing (NLP) perspective as a problem of binary classification in order to assess the correctness of a NLP system output. Previous work focusing on Machine Translation includes statistical methods for estimating correctness scores or correctness probabilities, following a rich search over the spectrum of possible features (**?**; **?**; **?**; **?**; **?**).

In this work we slightly transform the binary classification practice to fit the standard WMT human evaluation process. As human annotators have provided their evaluation in the form of ranking of five system outputs at a sentence level, we build our evaluation mechanism with similar functionality, aiming to training from and evaluating against this data. Evaluation scores and results can be then calculated based on comparative analysis of the performance of each system.

Whereas latest work, such as **?**), has focused on learning to assess segment performance independently for each system output, our contribution measures the performance by comparing the system outputs with each other and consequently ranking them. The exact method is described below.

2.2 Internal pairwise decomposition

We build one classifier over all input sentences. While the evaluation mechanism is trained and evaluated on a multi-class (ranking) basis as explained above, the classifier is expected to work on a binary level: we provide the features from the analysis of the two system outputs and the source, and the classifier should decide if the first system output is better than the second one or not.

In order to accomplish such training, the n systems' outputs for each sentence are broken down to $n \times (n-1)$ pairs, of all possible comparisons between two system outputs, in both directions (similar to the calculation of the Spearman correlation). For each pair, the classifier is trained with a class value c, for the pairwise comparison of system outputs t_i and t_j with respective ranks r_i and r_j, determined as:

$$c(r_i, r_j) = \begin{cases} 1 & r_i < r_j \\ -1 & r_i > r_j \end{cases}$$

At testing time, after the classifier has made all the pairwise decisions, those need to be converted back to ranks. System entries are ordered, according to how many times each of them won in the pairwise comparison, leading to rank lists similar to the ones provided by human annotators. Note that this kind of decomposition allows for *ties* when there are equal times of winnings.

2.3 Acquiring features

In order to obtain features indicating the quality of the MT output, automatic NLP analysis tools are applied on both the source and the two target (MT-generated) sentences of every pairwise comparison. Features considered can be seen in the following categories, according to their origin:

- **Sentence length:** Number of words of source and target sentences, source-length to target-length ratio.
- **Target language model:** Language models provide statistics concerning the correctness of the words' sequence on the target language. Such language model features include:
  - the smoothed n-gram probability of the entire target sentence for a language model of order 5, along with
  - uni-gram, bi-gram, tri-gram probabilities and a
  - count of unknown words
- **Parsing:** Processing features acquired from PCFG parsing (**?**) for both source and target side include:
  - parse log likelihood,
  - number of n-best trees,
  - confidence for the best parse,
  - average confidence of all trees.

  Ratios of the above target features to their respective source features were included.
- **Shallow grammatical match:** The number of occurences of particular node tags on both the source and the target was counted on the PCFG parses. In particular, NPs, VPs, PPs, NNs and punctuation occurences were counted. Then the ratio of the occurences of each tag in the target sentence by its occurences on the source sentence was also calculated.

2.4 Classifiers

The machine learning core of the system was built supporting two classification approaches.

- **Naïve Bayes** allows prediction of a binary class, given the assumption that the features are statistically independent.

$$p(C, F_1, \ldots, F_n) = p(C) \prod_{n}^{i=1} p(F_i|C)$$

$p(C)$ is estimated by relative frequencies of the training pairwise examples, while $p(F_i|C)$ for our continuous features are estimated with LOESS (locally weighted linear regression similar to **?**))

- **k-nearest neighbour** (knn) algorithm allows classifying based on the closest training examples in the feature space.

3 Experiment

3.1 Experiment setup

A basic experiment was designed in order to determine the exact setup and the feature set of the metric prior to the shared task submission. The classifiers for the task were learnt using the German-English testset of the WMT 2008 and 2010 (about 700 sentences)[1]. For testing, the classifiers were used to perform ranking on a test set of 184 sentences which had been kept apart from the 2010 data, with the criterion that they do not contain contradictions among human judgments.

In order to allow further comparison with other evaluation metrics, we performed an extended experiment: we trained the classifiers over the WMT 2008 and 2009 data and let them perform automatic ranking on the full WMT 2010 test set, this time without any restriction on human evaluation agreement.

In both experiments, tokenization was performed with the PUNKT tokenizer (**?**; **?**), while n-gram features were generated with the SRILM toolkit (**?**). The language model was relatively big and had been built upon all lowercased monolingual training sets for the WMT 2011 Shared Task, interpolated on the 2007 test set. As a PCFG parser, the Berkeley Parser (**?**) was preferred, due to the possibility of easily obtaining complex internal statistics, including n-best trees. Unfortunately, the time required for parsing leads to significant delays at the overall processing. The machine learning algorithms were implemented with the Orange toolkit (**?**).

3.2 Feature selection

Although the automatic NLP tools provided a lot of features (section 2.3), the classification methods we used (and particularly naïve Bayes were the development was focused on) would be expected to perform better given a smaller group of statistically independent features. Since exhaustive training/testing of all possible feature subsets was not possible, we performed feature selection based on the Relieff method (**?**; **?**). Automatic ranking was performed based on the most promising feature subsets. The results are examined below.

3.3 Results

The performance of the classifier is measured after the classifier output has been converted back to rank lists, similar to the WMT 2010 evaluation. We therefore calculated two types of rank coefficients: averaged Kendall's tau on a segment level, and Spearman's rho on a system level, based on the percentage that the each system's translations performed better than or equal to the translations of any other system.

The results for the various combinations of features and classifiers are depicted on Table 1. Naïve Bayes provides the best score on the test set, with $\rho = 0.81$ on a system level and $\tau = 0.26$ on a segment level, trained with features including the number of the unknown words, the source-length by target-length ratio, the VP count ratio and the source-target ratio of the parsing log-likelihood. The number of unknown words particularly appears to be a strong indicator for the quality of the sentence. On the first part of the table we can also observe that language model features do not perform as well as the features deriving from the processing information delivered by the parser. On the second part of the table we compare the use of various grammatical combinations. The third part contains the correlation obtained by various similar internal parsing-related features.

The correlation coefficients of the extended experiment, allowing comparison with last year's shared task, are shown on the last line of the table. With coefficients $\rho = 0.60$ and $\tau = 0.23$, our metric performs relatively low compared to the other metrics of WMT10 (indicatively iBLEU: $\rho = 0.95$, $\tau = 0.39$ according to **?**). Though, it still has a position in the list, scoring better than several other reference-aware metrics (e.g. of $\rho = 0.47$ and $\tau = 0.12$ respectively) for the particular language pair.

[1] data acquired from http://www.statmt.org/wmt11

features	naïve Bayes		knn	
	rho	tau	rho	tau
basic experiment				
ngram	0.19	0.05	0.13	0.01
unk, len	0.67	0.20	0.73	0.24
unk, len, bigram	0.61	0.21	0.74	0.21
unk, len, ngram	0.63	0.19	0.59	0.21
unk, len, trigram	0.67	0.20	0.76	0.21
unk, len, $\log_{parse}$	0.75	0.21	0.74	0.25
unk, len, n_{parse}, VP	0.67	0.24	0.61	0.20
unk, len, n_{parse}, VP, $conf_{bestparse}$	0.78	0.25	0.75	0.24
unk, len, n_{parse}, NP, $conf_{bestparse}$	0.78	0.23	0.74	0.23
unk, len, n_{parse}, VP, $conf_{avg}$	0.75	0.21	0.78	0.23
unk, len, n_{parse}, VP, $conf_{bestparse}$	0.78	0.25	0.75	0.24
unk, len, n_{parse}, VP, $\log_{parse}$	**0.81**	**0.26**	0.75	0.23
extended experiment				
unk, len, n_{parse}, VP, $\log_{parse}$	**0.60**	**0.23**	0.28	0.02

Table 1: System-level Spearman's rho and segment-level Kendall's tau correlation coefficients achieved on automatic ranking (average absolute value)

4 Discussion

A concern on the use of Confidence Estimation for MT evaluation has to do with the possibility of a system "tricking" such metrics. This would for example be the case when a system offers a well-formed candidate translation and gets a good score, despite having no relation to the source sentence in terms of meaning. We should note that we are not capable of fully investigating this case based on the current set of experiments, because all of the systems in our data sets have shown acceptable scores (11-25 BLEU and 0.58-0.78 TERp according to **?**)), when evaluated against reference translations. Though, we would assume that we partially address this problem by using ratios of source to target features (length, syntactic constituents), which means that in order for a sentence to trick the metric, it would need a comparable sentence length and a grammatical structure that would allow it to achieve feature ratios similar to the other systems' outputs. Previous work (**?**; **?**) has used features based on word alignment, such as IBM Models, which would be a meaningful addition from this aspect.

Although *k-nearest-neighbour* is considered to be a superior classifier, best results are obtained by naïve Bayes. This may have been due of the fact that feature selection has led to small sets of uncorrelated features, where naïve Bayes is known to perform well. *K-nearest-neighbour* and other complex classification methods are expected to prove useful when more complex feature sets are employed.

5 Conclusion and Further work

The experiments presented in this article indicate that confidence metrics trained over human rankings can be possibly used for several tasks of evaluation, given particular conditions, where e.g. there is no reference translation given. Features obtained from a PCFG parser seem to be leading to better correlations, given our basic test set. Although correlation is not particularly high, compared to other reference-aware metrics in WMT 10, there is clearly a potential for further improvement.

Nevertheless this is still a small-scale experiment, given the restricted data size and the single translation direction. The performance of the system on broader training and test sets will be evaluated in the future. Feature selection is also subject to change if other language pairs are introduced, while more sophisticated machine learning algorithms, allowing richer feature sets, may also lead to better results.

Acknowledgments

This work was done with the support of the TaraXŰ Project[2], financed by TSB Technologiestiftung Berlin–Zukunftsfonds Berlin, co-financed by the European Union–European fund for regional development.

[2] http://taraxu.dfki.de