Homework 4: Reranking

Roger Que

2014–04–17

This reranker is a simple implementation of the PRO pairwise ranking classification method [1]. For each source sentence in the training data, it computes a sentence-wise BLEU score for each candidate hypothesis against the reference sentence. It then samples pairs of hypotheses to build a relative ranking table according to these scores, and uses these samples to train a linear SVM. Optimization of the classifier's hyperparameter C is performed by 5-fold cross-validation. Finally, to select the best hypothesis for each sentence in the testing data, the reranker computes the dot product of the SVM's weight vector and each hypothesis' feature vector to derive a score.

Features for each hypothesis include:

- The translation and language model probabilities provided in the hypothesis data files.
- The word count (excluding punctuation) and number of verbs in each hypothesis, as counted by the TextBlob library [2]. The latter feature captures the intuition that translations with poor fluency may not have any verbs at all, and should be appropriately penalized.
- The number of words (again, excluding punctuation) that appear to have been left untranslated in the target sentence, computed by taking the cardinality of the intersection of the multisets of words in the source and target sentence. This simple implementation takes advantage of the fact that Russian and English, the two languages used in our data, use different scripts and thus should have near-zero word overlap.

Method	**Dev**	**Test**
All features (3)	28.23	28.11
With buggy gold scoring (4)	**28.93**	28.37
No verb count (1)	28.71	**28.45**

Table 1: Mean BLEU scores on development and testing data for different reranker settings. Numbers in parentheses indicate the number of runs averaged over. The best score for each data set is highlighted in bold.

The pairwise sampling algorithm used in PRO has two parameters, which are referred to as Γ and Ξ. For each source sentence, Γ pairwise samples are taken from the corresponding hypothesis set. Pairs with a gold score difference meeting a minimum threshold are added to a min-heap that stores up to Ξ pairs, effectively retaining the pairs with the highest score difference. In this case, the threshold is set to 0.05 BLEU, following that presented in the original paper. These heaps are finally combined and used to fit the classifier. Although the original implementation tuned Γ and Ξ on the training data, such a process yielded no significant difference in BLEU score on the development set for the values tested ($\Gamma \in \{10, 100, 1000\}$ and $\Xi \in \{1, 10, 100\}$), and so the values $\Gamma = \Xi = 100$ were arbitrarily selected.

The performance of the reranker with various settings, as measured by the corpus BLEU score, is shown in Table 1. For comparison, results on two non-final versions of the reranker are also shown. One does not implement the verb count feature. The other contains a buggy implementation of the

gold score computed over the whole line of the hypothesis data file, including the source sentence ID and probabilities, instead of only the actual text of the hypothesis.

Curiously, the full-featured version shows the worst performance on both the development and testing sets. The unexpectedly high mean performance of the implementation without verb counts on the testing data may be explained by the fact that only one run was submitted for evaluation; the buggy-gold-score version has the best single-run performance at 28.69 BLEU. In general, unfortunately, it is difficult to draw conclusive inferences on the value of each of the features tested from these results. This is due in part to the small size of the training set, which at only 400 sentences is unlikely to be large enough to yield feature weights that generalize well over unseen data. For such a small amount of data, a non-classifier-based method such as MERT or MIRA may present a better approach.

References

[1] Mark Hopkins and Jonathan May. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL: `http://www.aclweb.org/anthology/D11-1125`.

[2] Steven Loria and contributors. TextBlob: simplified text processing. URL: `http://textblob.readthedocs.org/`.