Homework 3: Evaluation

Roger Que

2014–03–27

This submission implements an SVM-based pairwise evaluator that ranks hypotheses according to three primary classes of features, or some subset thereof:

- word count;
- n-gram precision, recall, and F_1 score; and
- the compressibility of a document containing both the hypothesis and its corresponding reference.

In order to expand the usually binary classification of an SVM to cover all of the possible results of the ranking function $f(h_1, h_2, e) \in \{-1, 0, 1\}$, the evaluator uses the default "one-against-one" strategy of the underlying scikit-learn library [2]. A binary SVM is trained on each pair of labels. These results are then combined in order to produce a final label for each pair of hypotheses.

The word count and n-gram features were taken from the ROSE classifier, which also uses an SVM to score hypotheses [3].[1]

The last metric of compressibility is motivated by the information theoretic notion that the more similar two strings are, the smaller a compressed text containing both of them should be, controlling for each string's length [1]. This is because repeated substrings can be represented just once in the compressed text. For this feature, each hypothesis and its corresponding reference were concatenated together, then compressed using the gzip algorithm. The compressed lengths were then used as feature values.

Method	Train	Test
No n-gram features	79	49
Up to 1-grams	70	51
Up to 2-grams	66	51
Up to 3-grams	64	52
Up to 4-grams	62	52
No word count features	57	**53**
No n-grams, $C = 10^{-1}$	47	46
No n-grams, $C = 10^{0}$	79	49
No n-grams, $C = 10^{1}$	87	46
No n-grams, $C = 10^{2}$	**88**	46
Simple METEOR	51	—
Simple gzip size	50	—

Table 1: Percentage of correct rankings on training and testing data for various evaluation methods. The "simple" methods at the bottom of the table use pairwise ranking based on score comparisons instead of classification. The best percentage for each data set is highlighted in bold.

After the computation of feature values, the SVM hyperparameter C was selected from the values $\{10^{-2}, \ldots, 10^{3}\}$ using 5-fold cross-validation. In virtually all cases, this process yielded the value $C = 1$, indicating a balance between fitting to the training data (high C) and simplicity of the decision surface (low C).

A comparison of the accuracy of various classification methods on training and testing data is shown in Table 1. As expected, high values of C yield the best fit to the training data, but negatively

[1] The evaluator also implements a form of ROSE's "mixed" n-gram feature, which combines the text string and part of speech for each word. However, due to a coding oversight, a "hit" requires that both the string and POS match, unlike ROSE, which only requires one of the two to match. Thus this feature is not considered elsewhere in this writeup.

impact accuracy on testing data. Surprisingly, the addition of higher-order *n*-gram features has opposite effects between training and testing data. This is likely due to the lack of features causing overfitting on the remaining word count and compressibility information.

Classification without the word count feature yielded the best score on testing data, in spite of middling performance on the training set. This may be evidence of the impact on accuracy caused by the evaluator's failure to normalize the word count and compressed byte size features at training time. In the former case, this means that the word count features do not take the relative length of the reference into account, while in the latter, the feature vectors draw no distinction between sentences that are highly similar, and sentences that are simply shorter to begin with. An improved classifier should normalize these factors against the reference for each sentence, making the values more directly comparable across sentences.

References

[1] Marcus Dobrinkat, Jaakko Väyrynen, Tero Tapiovaara, and Kimmo Kettunen. Normalized compression distance based measures for MetricsMATR 2010. In *Proceedings of the joint fifth workshop on statistical machine translation and MetricsMATR*, pages 343–348. Association for Computational Linguistics, 2010.

[2] 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 E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[3] Xingyi Song and Trevor Cohn. Regression and ranking based optimisation for sentence level mt evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 123–129, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.