WLV: A confidence-based machine learning method for the GREC-NEG'09 task

Constantin Orăsan
RIILP
University of Wolverhampton, UK
C.Orasan@wlv.ac.uk

Iustin Dornescu
RIILP
University of Wolverhampton, UK
I.Dornescu@wlv.ac.uk

Abstract

This article presents the machine learning approach used by the University of Wolverhampton in the GREC-NEG'09 task. A classifier based on J48 decision tree and a meta-classifier were used to produce two runs. Evaluation on the development set shows that the meta-classifier achieves a better performance.

1 Introduction

The solution adopted by the University of Wolverhampton to solve the GREC-NEG task relies on machine learning. To this end, we assumed that it is possible to learn which is the correct form for a referential expression given the context in which it appears. The remainder of the paper is structured as follows: Section 2 presents the method used in this paper. Section 3 presents the evaluation results on the development set. The paper finishes with conclusions.

2 Method

The method used to solve the GREC-NEG task was inspired by the machine learning approaches employed for coreference resolution. In these methods, pairs of entities are classified as coreferential or not on the basis of a set of features (Mitkov, 2002). In the same manner, each REF element from the text to be processed is paired with all the REFEX elements in its chain and machine learning is used to determine the lexical form of which candidate REFEX element can be used in the given context. To achieve this, a set of features was derived after a corpus investigation. As can be seen, some of these features are similar to those used by resolution algorithms (e.g. distance between entities), whilst others are specific for the task (e.g. empty markers). The features used for a (REF, REFEX) pair are:

- Whether the REF element is the first mention in the chain. We noticed that in most cases it corresponds to the longest REFEX element in the *plain* case.
- Whether the REFEX element is the longest string.
- Whether the REF element is the first word in the sentence as this word is very likely to be the subject (i.e. *nominative* or *plain* case).
- Whether the words before the REF element can signal a possible empty element. Example of such phrases are ", but" and "and then". These phrases were extracted after analysing the training corpus.
- The distance in sentences to the previous REF element in the chain. This feature was used because a pronoun is more likely to be used when several mentions are in the same sentence, whilst full noun phrases are normally used if the mentions are far away or in different paragraphs.
- The REG08-TYPE of the REFEX tags that were assigned by the program to the previous 2 REF elements in the chain. This information can prove useful in conjunction with the previous feature.
- The part-of-speech tags of the four words before and three words after the REF element as a way to indicate the context in which the element appears.
- A compatibility feature which indicates pairs of SYNFUNC and CASE that are highly correlated. This correlation was determined by extracting the most frequent SYNFUNC and CASE pairs from the training corpus.

Proceedings of the 2009 Workshop on Language Generation and Summarisation, ACL-IJCNLP 2009, pages 107–108,
Suntec, Singapore, 6 August 2009.

- The size of the chain in elements as longer chains are more likely to contain pronouns.
- The values of SEMCAT, SYNCAT and SYNFUNC attributes of REF element and REG08-TYPE and CASE of REFEX element.
- The number of words in the REFEX value.
- Whether REF is in the first chain of the document.

The last two features were introduced in order to discriminate between candidate REFEX values that have the same *type* and *case*. For example, the number of words proved very useful when selecting genitive case names and chi-squared statistic ranks it as one of the best features together with the compatibility feature, information about previous elements in the chain and the longest REFEX candidate.

Before the features are calculated, the text is split into sentences and enriched with part-of-speech information using the OpenNLP library. [1] The instances are fed into a binary classifier that indicates whether the (REF, REFEX) pair is *good* (i.e. the REFEX element is a good filler for the REF element). Since each pair is classified independently, it is possible to have zero, one or more *good* REFEX candidates for a given REF. Therefore, the system uses the confidence returned by the classifier to rank the candidates and selects the one that has the highest probability of being *good*, regardless of the class assigned by the classifier. In this way the system selects exactly one REFEX for each REF.

3 Evaluation

The method proposed in this paper was evaluated using two classifiers, both trained on the same set of features. The first classifier is the standard J48 decision tree algorithm implemented in Weka (Witten and Frank, 2005). The run that used this classifier is referred to in the rest of the paper as *standard* run. Given the large number of negative examples present in our training data, a meta-classifier that is cost-sensitive was used for the second run. In our case, the meta-classifier relies on J48 and reweights training instances according to the total cost assigned to each class. After experimenting with different cost matrices, we decided to assign a cost of 3 to false negatives and 1 to false positives, in this way biasing the classifier towards a higher recall for YES answers. The results obtained using this meta-classifier are referred to as *biased* run. Our results on the development set are presented in Table 1.

Measure	Standard	Biased
classification accuracy	94.40%	92.09%
total pairs	907	907
reg08 type matches	621	728
reg08 type accuracy	68.46%	80.26%
reg08 type precision	68.46%	80.26%
reg08 type recall	66.20%	77.61%
string matches	568	667
string accuracy	62.62%	73.53%
mean edit distance	0.845	0.613
mean normalised edit distance	0.351	0.239

Table 1: The evaluation results on the development set

The first row in the table presents the accuracy of the classifier on the training data using 10-fold cross-validation. The very high accuracy is due to the large number of negative instances in the training data: assigning all the instances to the class NO achieves a baseline accuracy of 88.96%. The rest of the table presents the accuracy of the system on the development set using the script provided by the GREC-NEG organisers. As can be seen, the best results are obtained by the biased classifier despite performing worse at the level of classification accuracy. This can be explained by the fact that we do not use the output of the classifier directly, instead using the classification confidence.

4 Conclusions

This paper has presented our participation in the GREC-NEG task with a machine learning system. Currently the system tries to predict whether a (REF, REFEX) pair is valid, but in the future we plan to approach the task by using machine learning methods to determine the values of REG08-TYPE and CASE attributes.

References

Ruslan Mitkov. 2002. *Anaphora resolution*. Longman.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers.

[1] http://opennlp.sourceforge.net/