CNTS: Memory-Based Learning of Generating Repeated References

Iris Hendrickx, Walter Daelemans, Kim Luyckx, Roser Morante, Vincent Van Asch
CNTS, Department of Linguistics
University of Antwerp
Prinsstraat 13, 2000, Antwerp, Belgium
firstname.lastname@ua.ac.be

Abstract

In this paper we describe our machine learning approach to the generation of referring expressions. As our algorithm we use memory-based learning. Our results show that in case of predicting the TYPE of the expression, having one general classifier gives the best results. On the contrary, when predicting the full set of properties of an expression, a combined set of specialized classifiers for each subdomain gives the best performance.

1 Introduction

In this paper we describe the systems with which we participated in the GREC task of the REG 2008 challenge (Belz and Varges, 2007). The GREC task concerns predicting which expression is appropriate to refer to a particular discourse referent in a certain position in a text, given a set of alternative referring expressions for selection. The organizers provided the GREC corpus that consists of 2000 texts collected from Wikipedia, from 5 different subdomains (people, cities, countries, mountains and rivers) .

One of the main goals of the task is to discover what kind of information is useful in the input to make the decision between candidate referring expressions. We experimented with a pool of features and several machine learning algorithms in order to achieve this goal.

2 Method

We apply a standard machine learning approach to the task. We train a classifier to predict the correct label for each mention. As our machine learning algorithm we use memory-based learning as implemented in the Timbl package (Daelemans et al., 2007). To select the optimal algorithmic parameter setting for each classifier we used a heuristic optimization method called paramsearch (Van den Bosch, 2004). We also tried several other machine learning algorithms implemented in the Weka package (Witten and Frank, 2005), but these experiments did not lead to better results and are not further discussed here.

We developed four systems: a system that only predicts the TYPE of each expression (Type), so it predicts four class labels; and a system that predicts the four properties (TYPE, EMPATHIC, HEAD, CASE) of each expression simultaneously (Prop). The class labels predicted by this system are concatenated strings: 'common_no_nominal_plain', and these concatenations lead to 14 classes, which means that not all combinations appear in the training set. For both Type an Prop we created two variants: one general classifer (g) that is trained on all subdomains, and a set of combined specialized classifiers (s) that are optimized for each domain separately.

3 System description

To build the feature representations, we first preprocessed the texts performing the following actions: rule-based tokenization, memory-based part-of-speech tagging, NP-chunking, Named entity recognition, and grammatical relation finding (Daelemans and van den Bosch, 2005). We create an instance for each mention, using the following features to repre-

sent each instance:

- Positional features: the sentence number, the NP number, a boolean feature that indicates if the mention appears in the first sentence.
- Syntactic and semantic category given of the entity (SEMCAT, SYNCAT).
- Local context of 3 words and POS tags left and right of the entity.
- Distance to the previous mention measured in sentences and in NPs.
- Trigram pattern of the given syntactic categories of 3 previous mentions.
- Boolean feature indicating if the previous sentence contains another named entity than the entity in focus.
- the main verb of the sentence.

We do not use any information about the given set of alternative expressions except for post processing. In a few cases our classifier predicts a label that is not present in the set of alternatives. For those cases we choose the most frequent class label (as estimated on the training set).

We experimented with predicting all subdomains with the same classifier and with creating separate classifiers for each subdomains. We expected that semantically different domains would have different preferences for expressions.

4 Results

We provide results for the four systems Type-g, Type-s, Prop-g and Prop-s in Table 1. The evaluation script was provided by the organisers. The variant Type-g performs best with a score of 76.52% on the development set.

Data	Type-g	Type-s
Cities	64.65	60.61
Countries	75.00	71.74
Mountains	75.42	77.07
People	85.37	72.50
Rivers	65.00	80.00
All	**76.52**	72.26

Data	Prop-g	Prop-s
Cities	63.64	65.66
Countries	72.83	69.57
Mountains	72.08	74.58
People	79.51	79.51
Rivers	65.00	70.00
All	73.02	**73.93**

Table 1: Accuracy on GREC development set.

5 Conclusions

In this paper we described our machine learning approach to the generation of referring expressions. We reported results of four memory-based systems. Predicting all subdomains with the same classifier is more efficient when predicting the coarse-grained TYPE class. On the contrary, training specialized classifiers for each subdomain works better for the more fine-grained prediction of all properties simulaneously. For the test set we will present results the two best systems: CNTS-Type-g and CNTS-Prop-s.

Acknowledgments

This research is funded by FWO, IWT, GOA BOF UA, and the STEVIN programme funded by the Dutch and Flemish Governments.

References

A. Belz and S. Varges. 2007. Generation of repeated references to discourse entities. In *In Proceedings of the 11th European Workshop on Natural Language Generation (ENLG'07)*, pages 9–16.

W. Daelemans and A. van den Bosch. 2005. *Memory-based language processing*. Cambridge University Press, Cambridge, UK.

W. Daelemans, J. Zavrel, K. Van der Sloot, and A. Van den Bosch. 2007. TiMBL: Tilburg Memory Based Learner, version 6.1, reference manual. Technical Report 07-07, ILK, Tilburg University.

A. Van den Bosch. 2004. Wrapped progressive sampling search for optimizing learning algorithm parameters. In *Proceedings of the 16th Belgian-Dutch Conference on Artificial Intelligence*, pages 219–226.

I. H. Witten and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques, second edition*. Morgan Kaufmann, San Francisco.