Dutch Named Entity Recognition using Classifier Ensembles

Bart Desmet [a, b] Véronique Hoste [a, c]

[a] *LT3, Language and Translation Technology Team, University College Ghent*
[b] *Dept. of Applied Mathematics and Computer Science, Ghent University*
[c] *Dept. of Linguistics, Ghent University*

1 Introduction

Named Entity Recognition (NER) is the task of automatically identifying names within text and classifying them into categories, such as persons, locations and organizations. A variety of machine learning algorithms has been applied to the task, with research often aimed at feature selection and parameter optimization to improve a single classifier's performance. However, finding the optimal features and parameters is a complex problem.

An alternative research direction is to combine several classifiers into an ensemble, and combining their output using a voting procedure. The assumption is that combining a diverse set of classifiers improves the generalization accuracy, provided that the ensemble's members have sufficient individual performance and the errors they make are, to some extent, non-overlapping. Again, finding such an ensemble is a non-trivial problem.

In this paper, we describe a system that uses genetic algorithms to construct the best ensemble from a set of classifiers, based on the approach proposed in [3]. Instead of using only maximum entropy classifiers, however, we select classifiers from three different frameworks, namely memory-based learning (MBL), conditional random fields (CRF) and support vector machines (SVM), and evaluate the performance on a Dutch data set.

2 Experiments

We used a subset of the STEVIN-funded SoNaR corpus for our experiments, annotated with persons, organizations, locations, products and miscellaneous named entities [2].

In order to have a diverse pool of classifiers, 8 different feature sets were used to derive instance bases from the data set. The feature sets varied in the inclusion of typical features for NER, such as orthographic information (capitalization, punctuation, presence of digits, etc.), word shape and length, prefix and suffix, chunk tag and patterns (regexes for URLs and initials). Each set contained features for the original token, its POS tag, and its position in the sentence, because omitting them yielded classifiers that were deemed too weak for inclusion in an ensemble.

These feature sets were combined with 4 classification configurations, which were found to perform well and reasonably fast with all the features: TiMBL[1] with a k-value of 1 and a k-value of 7 (MBL), CRF++[2] and YamCha[3] (SVM). Two sets of MBL classifiers were used to have an equal amount of lazy and greedy learners, and varying the k-value yielded considerable differences in output, making the classifiers sufficiently diverse for potential inclusion in an ensemble. Each feature set was combined with every configuration to classify the dataset, using threefold cross-validation on all the data. This resulted in

[1] http://ilk.uvt.nl/timbl/
[2] http://crfpp.sourceforge.net/
[3] http://chasen.org/~taku/software/yamcha/

32 files with class tags for every token, to be used for the ensemble voting procedure. The best individual F-score (83.77 with threefold CV) was obtained by a CRF classifier.

When an ensemble of classifiers is used to determine the class of an instance, some sort of voting mechanism is needed to combine the class tags each individual in the ensemble has assigned to that instance. We experimented with four different voting systems: normal majority voting (each classifier casts an equally influential vote), and three forms of weighted voting where the influence of a classifier's vote was determined by its performance.

We used a genetic algorithm to approximate an optimal classifier ensemble. The genome consisted of a 32-bit string representing inclusion of each of the 32 classifiers. Fitness was determined by the F-score calculated on the voted output of the included classifiers. The genetic algorithm ran on a population size of 50, with rank selection, single point crossover with a probability of 0.90, flip mutation with a probability of 0.02 and 40 generations.

3 Results and discussion

The ensemble containing all classifiers obtained F-scores between 82.03 and 82.54, depending on the voting mechanism. The ensembles obtained with the genetic algorithm for each of the four voting mechanisms scored between 84.17 and 84.44. Scores are calculated on the voting output obtained from the individuals' threefold CV results on all the data.

A first observation that can be made is that the type of voting mechanism only has a minor influence on the performance of the best ensemble found by the genetic algorithm. The differences are somewhat more outspoken for the ensembles of all classifiers.

The four winning ensembles all contain classifiers from each of the three classification frameworks, although none of the TiMBL classifiers with k=7 is used. Two TiMBL classifiers with k=1 are present in one but all and all classifier ensembles, respectively. These classifiers achieve an individual F-score of 75.06 and 76.59, well below the F-scores of the selected CRF and SVM classifiers. This observation seems to corroborate that combining different types of learning algorithms in a classifier ensemble adds information, leading to better generalization performance of the ensemble.

All best-performing classifier ensembles outperform the ensembles consisting of all classifiers by a significant margin, and the difference in F-score between the best classifier ensemble (normal majority voting, 84.44) and the best individual classifier (CRF, 83.77) is 0.67 percentage points, a statistically significant difference. This confirms that genetic algorithms can be successfully applied to the task of finding a classifier ensemble that outperforms the best individual classifier.

However, the performance gain is not very large and comes at a high computational cost. This raises doubts about whether ensemble classification is a cheaper way to better performance than optimizing an individual classifier. In an additional experiment, the feature set of the best-performing CRF classifier was adapted to include the features of the second-best classifier it did not already have. This classifier achieved an F-score of 84.91 on the dataset, thus outperforming both the best individual classifier and the best ensemble classifier by 0.47 and 1.14 percentage points, respectively. This suggests that larger performance gains might be achieved if structural feature selection and parameter optimization would be applied [1]. Afterwards, classifier ensemble selection could be applied to optimize results even further.

References

[1] W. Daelemans, V. Hoste, F. De Meulder, and B. Naudts. Combined optimization of feature selection and algorithm parameters in machine learning of language. *Machine Learning,* 2837:84–95, 2003.

[2] B. Desmet and V. Hoste. Towards a balanced named entity corpus for Dutch. *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2010.

[3] A. Ekbal and S. Saha. Maximum entropy classifier ensembling using genetic algorithm for NER in Bengali. *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2010.