

## CHAPTER 9

---

### Conclusion

---

In this thesis, we presented a machine learning approach to the resolution of coreferential relations between nominal constituents in Dutch. It is the first automatic resolution approach proposed for this language. In order to enable a corpus-based strategy, we first annotated a corpus of Dutch news magazine text, KNACK-2002, with coreferential information for pronominal, proper noun and common noun coreferences. A separate learning module was built for each of these NP types. The main motivation for this approach was that the information sources which are important for the resolution of the coreferential relations differ per NP type. This approach was not only applied to Dutch, for which no comparative results are yet available, but also to the well-known English MUC-6 and MUC-7 data sets.

Coreference and the task of coreference resolution was the main point of interest in Chapters 2 and 3 and in Chapter 8 on testing. In the chapters in between, we focused on the methodological issues which arise when performing a machine learning of coreference resolution experiment, or more broadly, a machine learning of language experiment. In the following two sections, we discuss the main observations from the research questions formulated in Section 1.3.

## 9.1 Methodological issues: main observations

**Algorithm ‘bias’** In Chapter 4 we investigated the effect of algorithm ‘bias’ on learning coreference resolution. This was done because to our knowledge, this effect of ‘bias’ was not yet systematically investigated in the existing machine learning of coreference resolution literature. The existing machine learning approaches to coreference resolution use the C4.5 decision tree learner (Quinlan 1993), used by Aone and Bennett (1995), McCarthy (1996) and Soon et al. (2001), maximum entropy learning as in Yang et al. (2003) or the RIPPER rule learner (Cohen 1995) as in Ng and Cardie (2002a;2002b;2002c). By contrasting the performance of two completely different learning techniques, namely memory-based learning and rule induction, on this task of coreference resolution, we wanted to determine the effect of algorithm ‘bias’ on learning coreference resolution. Two machine learning packages were used in the experiments: the memory-based learning package TIMBL (Daelemans et al. 2002) and the rule induction package RIPPER (Cohen 1995). Independently of the type of data set, some clear tendencies could be observed with respect to precision and recall scores (Table 4.3). The precision scores for TIMBL were up to about 30% lower than the ones for RIPPER. For the recall scores, the opposite tendency could be observed, but to a lesser degree: TIMBL generally obtains a higher recall than RIPPER. Based on these tendencies we formulated some conclusions in terms of ‘bias’:

- With respect to the large difference in precision scores, we hypothesized that this was mainly due to the different feature handling in both learning techniques: RIPPER uses embedded feature selection for the construction of its rules, whereas TIMBL performs feature weighting, without taking into account the dependencies between features. One implication of this use of feature weighting is that a large group of features with low informativeness can overrule more informative features. This hypothesis was further investigated in Chapter 5.
- Furthermore, with respect to the lower recall scores for RIPPER, we hypothesized that the rule induction approach was more sensitive to the skewed class distribution in our data sets. In a lazy learning approach, all instances are stored in memory and no attempt is made to simplify the model by eliminating low frequency events, whereas in a eager learning approach such as RIPPER, possibly interesting information from the training data is either thrown away by pruning or made inaccessible by the eager construction of the model. For our data sets, this implies that RIPPER prunes away possibly interesting low-frequency positive data, which is harmful for its recall scores. This hypothesis was further investigated in Chapter 7.

With respect to the use of three classifiers trained on a different type of coreferential NPs, instead of one single classifier, we could observe that the RIPPER results of the three classifiers were always higher than the single classifier results, whereas the TIMBL results of the three classifiers were similar or even significantly below the single classifier results.

**Feature selection** Although the search for disambiguating features is central in the machine learning research for coreference resolution and for NLP tasks in general, the importance of feature selection has only recently been systematically investigated, as in Soon et al. (2001) and Ng and Cardie (2002c). In our experiments reported in Chapter 5, we opted for a more systematic and verifiable feature selection approach. We used three automated techniques for the selection of the relevant features, viz. backward elimination, bidirectional hillclimbing and a genetic algorithm. These three approaches start the search at a different starting point, when searching the space of feature subsets. The main objective was to determine the effect of feature selection on classifier performance. For TIMBL, we hypothesized that feature selection would lead to an increase of the precision scores. Feature selection indeed lifted the precision scores for TIMBL with up to 35% (Table 5.4, 5.5). As expected, this increase was much smaller (always less than 4%) for RIPPER due to the embedded feature selection used for the construction of the rules.

In these experiments, we also investigated whether the information sources considered to be optimal for one learner could be generalized to the other learner. With respect to the selected features, we observed that no general conclusions could be drawn (e.g. Table 5.6). Per language, per NP type data set (pronouns, proper nouns and common nouns) and per selection procedure, a different feature combination was selected by each learning algorithm. We concluded that the optimal feature selection had to be determined experimentally for each single data set. We consider this a rather disappointing result since this implies that the importance of the information sources cannot be considered as an isolated phenomenon. We were not able to determine a global set of features which holds for the task of coreference resolution. The whole experimental context with factors such as algorithm bias, algorithm parameters (Chapter 5) and class distribution (Chapter 7) interacts with the selection of information sources.

**Parameter optimization** In Chapter 5, we investigated the effect of parameter optimization on classifier performance. The main motivation for these experiments was that although most learning systems provide sensible default settings, it is by no means certain that they will be *optimal* parameter settings for some particular task. We performed an exhaustive variation of a number

of TIMBL and RIPPER parameters. Although the badly performing parameter combinations were in the minority, all experiments (Table 5.2, 5.3) revealed a lot of variation in the  $F_{\beta=1}$  results when varying the algorithm parameters. We observed that the method-internal performance differences could be much larger than the method-comparing performance differences. For both learners we could conclude that parameter optimization overall leads to large performance increases (Table 5.7, 5.8).

In the parameter optimization experiments, we again investigated whether the optimal parameters for a given learning method could be generalized to the different data sets. However, no general conclusion could be drawn concerning these settings (Table 5.7, 5.8). The optimal settings merely revealed some tendencies, such as the predominant use of MVDM (Modified Value Difference Metric) and weighted voting for TIMBL, and the above average use of minimal description length instance ordering and a below zero loss ratio value for RIPPER. This predominant use of MVDM has also been observed in the experiments investigating the effect of the interaction of feature selection and parameter optimization. In fact, through the use of MVDM, a combination of supervised and unsupervised learning is obtained. This metric can be considered as a clustering approach in which similar feature values are grouped in clusters which are relevant for the task. Also for other NLP tasks (Buchholz 2002), this metric has been shown to perform well.

In a next optimization step, we investigated if the above described information sources and algorithm parameters also interact. These experiments were conducted since there appears to be little understanding in the current literature of the interaction between these variables. In case optimization is performed, this is mostly done sequentially, which may not be not advisable if different experimental factors interact.

**Interaction of feature selection and parameter optimization** In Chapter 6, we investigated the effect of the interaction of feature selection and parameter optimization. We used a genetic algorithm as a feasible method to do this costly optimization. The GA optimization experiments confirm the tendencies observed in the isolated feature selection and parameter optimization experiments (Table 6.1):

Feature selection, parameter optimization and their joint optimization can cause large variation in the results of both classifiers. All three optimization steps lead to a large improvement over the default results. Furthermore, optimization mainly wipes out the initial weaknesses of TIMBL and RIPPER in their default settings: the increase of  $F_{\beta=1}$  scores for TIMBL is mainly obtained through a large increase of precision scores for TIMBL, whereas the increase of  $F_{\beta=1}$  scores

for RIPPER is mainly due to the increase of recall scores. Furthermore, we could once again observe that the performance differences inside one single learning method could be much larger than the method-comparing performance differences. Also, the optimization results did not reveal a clear supremacy of one learner over the other, which once again confirms the necessity of optimization.

With respect to the use of three classifiers, each trained on the coreferential relations of a specific type of NP, instead of one single classifier covering all coreferential relations, the following could be observed. Three classifiers, each trained on one specific NP type, perform better than one single classifier in 5 out of 6 data sets. But since this difference in performance is only significant in half of the cases, we concluded that no convincing evidence was found for our initial hypothesis that three more specialized classifiers, each trained on the coreferential relations of a specific type of NP would perform better on the task of coreference resolution than one single classifier covering all coreferential relations.

We also investigated whether general conclusions could be drawn with respect to the selected features and optimal parameters.

- The following observations could be made with respect to the selected features: RIPPER selects fewer features than TIMBL, which can be explained through the different feature handling in both learning techniques. For RIPPER, a feature is either on or off. For TIMBL, a feature is either on, off or MVDM and it also incorporates different feature weighting techniques to assign different degrees of informativeness to the selected features.

Furthermore, with respect to the informativeness of the features, we could observe that all features are informative for our task of coreference resolution. This observation refines the results displayed in Table 5.1 and also those reported by Soon et al. (2001), which show the lack of informativeness of the majority of the features, when they are considered in isolation. Furthermore, we could also observe that the initial predominance of the string-matching features (as also observed by Soon et al. (2001), Yang et al. (2004b) and others) has disappeared in favour of a more balanced combination of features.

A last observation made with respect to the selected features, was that the feature selection considered to be optimal for TIMBL could be different from the one optimal for RIPPER. TIMBL and RIPPER often incorporate different features in their instances (see for example Figure 6.3).

- Although the parameter settings which were selected after optimization could not be generalized, not even within one single data set and although the parameter settings that were optimal when using all features were

not necessarily optimal when performing feature selection, some general observations could be made.

For TIMBL, we could see that 99% of all optimal individuals consisted of a combination of features for which the distance calculation is handled by the overlap metric and features handled by the MVDM metric. With respect to the *distance weights*, we could observe that the different distance weighted class voting schemes were preferred above the default majority voting (9%). Furthermore, 97% of the different selected values of  $k$  was higher than the default  $k=1$ , which could be explained through the use of the MVDM metric in nearly all optimal individuals.<sup>1</sup>

For RIPPER, the most noticeable observation was made with respect to the *loss ratio* parameter, which allows to change the ratio of the cost of a false negative to the cost of a false positive. The default value of 1 was selected in only 3% of the cases, whereas all other individuals had a loss ratio value below 1 which implies that more importance is given to an improvement of the recall. This focus of recall can be explained through the skewedness of the data and the sensitivity of RIPPER to this skewedness (as investigated in Chapter 7). Since the positive class only represents a small fraction of the data, a large number of errors is made on the positive minority class. By decreasing the loss ratio value, an improvement on the recall scores can be obtained. Another clear observation was that the ordering method in which the classes are ordered by increasing frequency was selected in two thirds of the individuals (78%), whereas the ordering method which orders the classes by decreasing frequency was never selected. This parameter selection choice can again be explained through the skewed class distribution. For such a data set, an ordering method in which the classes are ordered by increasing frequency, makes more sense. This implies that first rules are learned for the positive minority class, whereas the negative class is taken as default classification. With respect to the number of *optimization passes* taken over the rules RIPPER learns, the default value 2 was selected in 91% of the individuals.

**Class distribution** In Chapter 7, we investigated how the class distribution of the data affects learning. In order to investigate the effect of class distribution on the performance of TIMBL and RIPPER, we created a variety of class distributions through the use of down-sampling and by changing the loss ratio parameter in RIPPER. For the down-sampling experiments we could conclude for the two learning methods that a decreasing rate of negative instances was beneficial for

---

<sup>1</sup>The MVDM metric groups feature values by looking at co-occurrence of values with target classes; this implies that the nearest neighbour set will usually be much smaller for MVDM than for the overlap metric at the same value of  $k$ .

recall. The same conclusion could be drawn in the experiments in which the loss ratio parameter was varied for RIPPER. Another general conclusion was that both down-sampling and a change of the loss ratio parameter below 1 was harmful for precision. We also showed that both learning approaches behave quite differently in case of skewedness of the classes and that they also react differently to a change in class distribution. TIMBL, which performs better on the minority class than RIPPER in case of a largely imbalanced class distribution, mainly suffers from a rebalancing of the data set. In contrast, the RIPPER results are sensitive to a change of class distribution or loss ratio. A decrease of the number of negative instances counters this pruning.

All these observations, however, are not limited to the task of coreference resolution. In earlier work (Hoste et al. 2002, Daelemans and Hoste 2002, Daelemans, Hoste, De Meulder and Naudts 2003, Decadt et al. 2004), we came to similar conclusions for the task of word sense disambiguation, the prediction of diminutive suffixes and part-of-speech tagging and for some non-NLP data sets.

In a typical comparative machine learning of language experiment, the impact of the factors discussed here is too often underestimated. In most comparative machine learning experiments, at least in computational linguistics, two or more algorithms are compared for a fixed sample selection, feature selection, feature representation, and (default) algorithm parameter setting over a number of trials (cross-validation), and if the measured differences are statistically significant, conclusions are drawn about which algorithm is better suited and why (mostly in terms of algorithm bias). Sometimes different sample sizes are used to provide a learning curve, and sometimes a limited parameter optimization is performed. No overall optimization of parameters, architecture and feature representation is undertaken (e.g. Mooney (1996), Escudero et al. (2000), Ng and Lee (1996), Lee and Ng (2002)). These studies explore only a few points in the space of possible experiments for each algorithm to be compared.

This methodology has already been criticized by Banko and Brill (2001), who showed that increasing the data sample size can strongly affect comparative results. In this study, we showed that changing any of the architectural variables, such as algorithm parameters, information sources and class distribution, can have great effects on the performance of a learning method, making questionable ‘hard’ conclusions in the literature on the relative adequacy of machine learning methods for a given task or on the importance of the information sources for solving a task, based on default settings of algorithms or on limited optimization only. Our experiments showed that there is a high risk that other areas in the experimental space may lead to radically different results and conclusions. In general, we conclude that the more effort is put in optimization, through feature selection, parameter optimization, sample selection and their joint optimization, the more reliable the results and the comparison will be.

## 9.2 Future research goals

Our future research goals relate to the observations made in Chapter 8. In this chapter, we showed that the results obtained on the MUC-6 (TIMBL: 64.3% and RIPPER: 63.4%) and MUC-7 (TIMBL: 60.2% and RIPPER: 57.6%) data sets were comparable to the results reported by Soon et al. (2001). Although the best results reported to date on the MUC-6 and MUC-7 data (Ng and Cardie 2002a, Ng and Cardie 2002b, Ng and Cardie 2002c) are significantly higher (69.5% on MUC-6 and 63.4%  $F_\beta$  on MUC-7), the field of coreference resolution still presents some major challenges. Furthermore, the  $F_\beta$  score of 51% of both TIMBL and RIPPER on the Dutch data showed that coreference resolution for Dutch is even more challenging.

Although we cannot quantify the error load of the different types of errors, since this would require a complete analysis of the different test corpora, we could get an impression of the major sources of errors through a qualitative error analysis of three English and three Dutch texts. We will now discuss the observations made for the different types of NPs and discuss some directions for future research.

With respect to *pronominal coreference*, we observed for both languages that a large part of the errors made by the pronominal resolution system involves the false distinction between anaphoric and pleonastic pronouns. Therefore, more effort should be put in features which can capture this difference. Another possible approach is to train a classifier, as in Mitkov et al. (2002), which automatically classifies instances of “it” as pleonastic or nominal anaphora.

For Dutch, the resolution of the pronominal anaphors is also severely hindered by part-of-speech tagging errors (e.g. the female “ze” is often erroneously tagged as a third person plural noun and vice versa). Since preprocessing errors are also a major source of errors for the Dutch proper noun and common noun resolution, we must conclude that the shallow parser trained on the Spoken Dutch Corpus is not suitable for this type of corpus. Therefore, we conclude that the whole part-of-speech tagging, NP chunking and relation finding procedure for Dutch should be reconsidered.

Furthermore, for the Dutch male and female pronouns, such as “hij”, “hem”, “zijn”, “haar”, we saw that the search space of candidate antecedents is much larger than that for the corresponding English pronouns, since they can also refer to the linguistic gender of their antecedent. The current feature vectors describing this type of relation have a low informativeness. Therefore, for the resolution of anaphors referring to the linguistic gender of their antecedent other features should be considered.

With respect to *proper noun coreference*, high precision scores ranging between 78.0% and 83.0% could be observed over all data sets. For both Dutch and English, the errors on the proper nouns are mainly caused by preprocessing errors: errors in NP chunking, part-of-speech tagging, relation finding, named entity recognition, apposition detection and alias detection. Therefore, more attention should be given to each of these preprocessing steps.

With respect to *common noun coreference*, we could observe low precision scores ranging between 57.2% and 47.5% on the three data sets. A similar observation was made by Ng and Cardie (2002a) and Strube et al. (2002) (for German). As for coreference resolution of anaphoric proper nouns, errors were caused by preprocessing errors, such as part-of-speech tagging, NP chunking, apposition recognition, etc. Other errors, such as the lack of detecting synonyms, hypernyms and paraphrases, are typical for the resolution of coreferential relations between common nouns. For this type of coreferential relations a large amount of semantic and world knowledge is required. In the construction of the instances for the English and Dutch data, we used WordNet and the Dutch EuroWordNet to build a set of semantic features. But both lexical resources, and in particular the Dutch EuroWordNet are restricted and miss a lot of commonly used expressions and their lexical relations. Furthermore, a lot of coreferential relations between NPs are restricted in time, such as the pair “Chirac”-“the president of France”, or names of political parties (e.g. “VLD”-“de Vlaamse liberalen”, “Agalev”- “de groenen”). In order to overcome the lack of information in the existing resources and in order to capture “dynamic” coreferential relations, we plan to use the Web as a resource (as for example Keller, Lapata and Ourioupina (2002), Turney (2001), Modjeska, Markert and Nissim (2003), and Bunescu (2003)).

