Proceedings of the 2002 Conference on Empirical Methods in
Natural Language Processing (EMNLP), Philadelphia, PA, July 2002, pp. 55-62.

Combining Sample Selection and Error-Driven Pruning for Machine Learning of Coreference Rules

Vincent Ng and **Claire Cardie**
Department of Computer Science
Cornell University
Ithaca, NY 14853-7501
{yung,cardie}@cs.cornell.edu

Abstract

Most machine learning solutions to noun phrase coreference resolution recast the problem as a classification task. We examine three potential problems with this reformulation, namely, skewed class distributions, the inclusion of "hard" training instances, and the loss of transitivity inherent in the original coreference relation. We show how these problems can be handled via intelligent sample selection and error-driven pruning of classification rulesets. The resulting system achieves an F-measure of 69.5 and 63.4 on the MUC-6 and MUC-7 coreference resolution data sets, respectively, surpassing the performance of the best MUC-6 and MUC-7 coreference systems. In particular, the system outperforms the best-performing learning-based coreference system to date.

1 Introduction

Noun phrase coreference resolution refers to the problem of determining which noun phrases (NPs) refer to each real-world entity mentioned in a document. Machine learning approaches to this problem have been reasonably successful, operating primarily by recasting the problem as a *classification task* (e.g. Aone and Bennett (1995), McCarthy and Lehnert (1995), Soon et al. (2001)). Specifically, an inductive learning algorithm is used to train a classifier that decides whether or not two NPs in a document are coreferent. Training data are typically created by relying on coreference chains from the training documents: training instances are generated by pairing each NP with each of its preceding NPs; instances are labeled as *positive* if the two NPs are in the same coreference chain, and labeled as *negative* otherwise.[1]

A separate clustering mechanism then coordinates the possibly contradictory pairwise coreference classification decisions and constructs a partition on the set of NPs with one cluster for each set of coreferent NPs. Although, in principle, any clustering algorithm can be used, most previous work uses a single-link clustering algorithm to impose coreference partitions.[2] An implicit assumption in the choice of the single-link clustering algorithm is that coreference resolution is viewed as anaphora resolution, i.e. the goal during clustering is to find an antecedent for each anaphoric NP in a document.[3]

Three intrinsic properties of coreference[4], however, make the formulation of the problem as a classification-based single-link clustering task potentially undesirable:

Coreference is a rare relation. That is, most NP pairs in a document are not coreferent. Con-

[1] Two NPs are in the same coreference chain if and only if they are coreferent.

[2] One exception is Kehler's work on probabilistic coreference (Kehler, 1997), in which he applies *Dempster's Rule of Combination* (Dempster, 1968) to combine all pairwise probabilities of coreference to form a partition.

[3] In this paper, we consider an NP anaphoric if it is part of a coreference chain but is not the head of the chain.

[4] Here, we use the term *coreference* loosely to refer to either the problem or the binary relation defined on a set of NPs. The particular choice should be clear from the context.

sequently, generating training instances by pairing each NP with each of its preceding NPs creates highly skewed class distributions, in which the number of positive instances is overwhelmed by the number of negative instances. For example, the standard MUC-6 and MUC-7 (1995; 1998) coreference data sets contain only 2% positive instances. Unfortunately, learning in the presence of such skewed class distributions remains an open area of research in the machine learning community (e.g. Pazzani et al. (1994), Fawcett (1996), Cardie and Howe (1997), Kubat and Matwin (1997)).

Coreference is a discourse-level problem with different solutions for different types of NPs. The interpretation of a pronoun, for example, may be dependent only on its closest antecedent and not on the rest of the members of the same coreference chain. Proper name resolution, on the other hand, may be better served by ignoring locality constraints altogether and relying on string-matching or more sophisticated aliasing techniques. Consequently, generating positive instances from all pairs of NPs from the same coreference chain can potentially make the learning task harder: all but a few coreference links derived from any chain might be hard to identify based on the available contextual cues.

Coreference is an equivalence relation. Recasting the problem as a classification task precludes enforcement of the transitivity constraint. After training, for example, the classifier might determine that A is coreferent with B, and B with C, but that A and C are not coreferent. Hence, the clustering mechanism is needed to coordinate these possibly contradictory pairwise classifications. In addition, because the coreference classifiers are trained independent of the clustering algorithm to be used, improvements in classification accuracy do not guarantee corresponding improvements in clustering-level accuracy, i.e. overall performance on the coreference resolution task might not improve.

This paper examines each of the above issues. First, to address the problem of skewed class distributions, we apply a technique for negative instance selection similar to that proposed in Soon et al. (2001). In contrast to results reported there, however, we show empirically that system performance increases noticeably in response to negative example selection, with increases in F-measure of 3-5%.

Second, in an attempt to avoid the inclusion of "hard" training instances, we present a corpus-based method for implicit selection of positive instances. The approach is a fully automated variant of the example selection algorithm introduced in Harabagiu et al. (2001). With positive example selection, system performance (F-measure) again increases, by 12-14%.

Finally, to more tightly tie the classification- and clustering-level coreference decisions, we propose an error-driven rule pruning algorithm that optimizes the coreference classifier ruleset with respect to the clustering-level coreference scoring function. Overall, the use of pruning boosts system performance from an F-measure of 69.3 to 69.5, and from 57.2 to 63.4 for the MUC-6 and MUC-7 data sets, respectively, enabling the system to achieve performance that surpasses that of the best MUC coreference systems by 4.6% and 1.6%. In particular, the system outperforms the best-performing learning-based coreference system (Soon et al., 2001) by 6.9% and 3.0%.

The remainder of the paper is organized as follows. In sections 2 and 3, we present the machine learning framework underlying the baseline coreference system and examine the effect of negative sample selection. Section 4 presents our corpus-based algorithm for selection of positive instances. Section 5 describes and evaluates the error-driven pruning algorithm. We conclude with future work in section 6.

2 The Machine Learning Framework for Coreference Resolution

Our machine learning framework for coreference resolution is a standard combination of classification and clustering, as described above.

Creating an instance. An instance in our machine learning framework is a description of two NPs in a document. More formally, let NP_{kd} be the kth NP in document d. An instance formed from NP_{id} and NP_{jd} is denoted by $i_{(NP_{id},NP_{jd})}$. A *valid* instance is an instance $i_{(NP_{id},NP_{jd})}$ such that NP_{id} precedes NP_{jd}.[5] Following previous work (Aone and Bennett (1995),

[5]By definition, exactly $\binom{n}{2}$ valid instances can be created from n NPs in a given document.

Soon et al. (2001)), we assume throughout the paper that only valid instances will be generated and used for training and testing. Each instance consists of 25 features, which are described in Table 1.[6] The classification associated with a training instance is one of COREFERENT or NOT COREFERENT depending on whether the NPs co-refer in the associated training text.[7]

Building an NP coreference classifier. We use RIPPER (Cohen, 1995), an information gain-based propositional rule learning system, to train a classifier that, given a test instance $i_{(NP_{id},NP_{jd})}$, decides whether or not NP_{id} and NP_{jd} are coreferent. Specifically, RIPPER sequentially covers the positive training instances and induces a ruleset that determines when two NPs are coreferent. When none of the rules in the ruleset is applicable to a given NP pair, a default rule that classifies the pair as not coreferent is automatically invoked. The output of the classifier is either COREFERENT or NOT COREFERENT along with a number between 0 and 1 that indicates the confidence of the classification.

Applying the classifier to create coreference chains. After training, the resulting ruleset is used by a best-first clustering algorithm to impose a partitioning on all NPs in the test texts, creating one cluster for each set of coreferent NPs. Texts are processed from left to right. Each NP encountered, NP_{jd}, is compared in turn to each preceding NP, NP_{id}, from right to left. For each pair, a test instance is created as during training and is presented to the coreference classifier. The NP with the highest confidence value among the preceding NPs that are classified as being coreferent with NP_{jd} is selected as the antecedent of NP_{jd}; otherwise, no antecedent is selected for NP_{jd}.

3 Negative Sample Selection

As noted above, skewed class distributions arise when generating all valid instances from the training texts. A number of methods for handling skewed distributions have been proposed in the machine learning literature, most of which modify the learning algorithm to incorporate a loss function with a much larger penalty for minority class errors than for instances from the majority classes (e.g. Gordon and Perlis (1989), Pazzani et al. (1994)). We investigate here a different approach to handling skewed class distributions — negative sample selection, i.e. the selection of a smaller subset of negative instances from the set of available negative instances. In the case of NP coreference, we hypothesize that reducing the number of negative instances will improve recall but potentially reduce precision: intuitively, the existence of fewer negative instances should allow RIPPER to more liberally induce positive rules. We propose a method for negative sample selection that, for each anaphoric NP, NP_{jd}, retains only those negative instances for non-coreferent NPs that lie between NP_{jd} and its **farthest** preceding antecedent, $f(\text{NP}_{jd})$. The algorithm for negative sample selection, NEG-SELECT, is shown in Figure 1. NEG-SELECT takes as input the set of all possible negative instances in the training texts, i.e. the set of valid instances $i_{(NP_{id},NP_{jd})}$ such that NP_{id} and NP_{jd} are not in the same coreference chain.

Algorithm NEG-SELECT(NEG: set of all possible negative instances)
for $i_{(NP_{id},NP_{jd})} \in$ NEG **do**
  if NP_{jd} is anaphoric **then**
    if NP_{id} precedes $f(\text{NP}_{jd})$ **then**
      NEG := NEG $\setminus \{i_{(NP_{id},NP_{jd})}\}$
  else
    NEG := NEG $\setminus \{i_{(NP_{id},NP_{jd})}\}$
return NEG

Figure 1: The NEG-SELECT algorithm

The intuition behind this approach is very simple. Let $S(\text{NP}_{jd})$ be the set of preceding antecedents of NP_{jd}, and $L(\text{NP}_{id},\text{NP}_{jd})$ be the set consisting of NPs NP_{id}, $\text{NP}_{(i+1)d}, \ldots, \text{NP}_{jd}$. Recall that the goal during clustering is to compute, for each NP NP_{jd}, the set $S(\text{NP}_{jd})$ from which the element with the highest confidence is selected as the antecedent of NP_{jd}. Since (1) $S(\text{NP}_{jd})$ is a subset of $L(f(\text{NP}_{jd}),\text{NP}_{jd})$[8] and

[6]See Ng and Cardie (2002) for a detailed description of the features.

[7]In all of the work presented here, NPs are identified, and feature values computed entirely automatically.

[8]We define $L(f(\text{NP}_{jd}),\text{NP}_{jd})$ to be the empty set if $f(\text{NP}_{jd})$ does not exist (i.e. NP_{jd} is not anaphoric).

Feature Type	Feature	Description
Lexical	PRO_STR	C if both NPs are pronominal and are the same string; else I.
	PN_STR	C if both NPs are proper names and are the same string; else I.
	SOON_STR_NONPRO	C if both NPs are non-pronominal and the string of NP_{id} matches that of NP_{jd}; else I.
Grammatical	PRONOUN_1	Y if NP_{id} is a pronoun; else N.
	PRONOUN_2	Y if NP_{jd} is a pronoun; else N.
	DEMONSTRATIVE_2	Y if NP_{jd} starts with a demonstrative such as "this," "that," "these," or "those;" else N.
	BOTH_PROPER_NOUNS	C if both NPs are proper names; NA if exactly one NP is a proper name; else I.
	NUMBER	C if the NP pair agree in number; I if they disagree; NA if number information for one or both NPs cannot be determined.
	GENDER	C if the NP pair agree in gender; I if they disagree; NA if gender information for one or both NPs cannot be determined.
	ANIMACY	C if the NPs match in animacy; else I.
	APPOSITIVE	C if the NPs are in an appositive relationship; else I.
	PREDNOM	C if the NPs form a predicate nominal construction; else I.
	BINDING	I if the NPs violate conditions B or C of the Binding Theory; else C.
	CONTRAINDICES	I if the NPs cannot be co-indexed based on simple heuristics; else C. For instance, two non-pronominal NPs separated by a preposition cannot be co-indexed.
	SPAN	I if one NP spans the other; else C.
	MAXIMALNP	I if both NPs have the same maximal NP projection; else C.
	SYNTAX	I if the NPs have incompatible values for the BINDING, CONTRAINDICES, SPAN or MAXIMALNP constraints; else C.
	INDEFINITE	I if NP_{jd} is an indefinite and not appositive; else C.
	PRONOUN	I if NP_{id} is a pronoun and NP_{jd} is not; else C.
	EMBEDDED_1	Y if NP_{id} is an embedded noun; else N.
	TITLE	I if one or both of the NPs is a title; else C.
Semantic	WNCLASS	C if the NPs have the same WordNet semantic class; I if they don't; NA if the semantic class information for one or both NPs cannot be determined.
	ALIAS	C if one NP is an alias of the other; else I.
Positional	SENTNUM	Distance between the NPs in terms of the number of sentences.
Others	PRO_RESOLVE	C if NP_{jd} is a pronoun and NP_{id} is its antecedent according to a naive pronoun resolution algorithm; else I.

Table 1: Feature Set for the Coreference System. The feature set contains relational and non-relational features. Non-relational features test some property P of one of the NPs under consideration and take on a value of **Y**ES or **N**O depending on whether P holds. Relational features test whether some property P holds for the NP pair under consideration and indicate whether the NPs are **C**OMPATIBLE or **I**NCOMPATIBLE w.r.t. P; a value of **N**OT **A**PPLICABLE is used when property P does not apply.

(2) NP_{jd} is compared to each preceding NP from *right* to *left* by the clustering algorithm, it follows that the set of negative instances whose classifications the classifier needs to determine in order to compute $S(\text{NP}_{jd})$ is a superset of the set of instances $I(\text{NP}_{jd})$ formed by pairing NP_{jd} with each of its non-coreferent preceding NPs in $L(f(\text{NP}_{jd}),\text{NP}_{jd})$. Consequently, $\bigcup_{j,d} I(\text{NP}_{jd})$ is the **minimal** set of (negative) instances whose classifications will be required during clustering. In principle, to perform the classifications accurately, the classifier needs to be trained on the corresponding set of negative instances from the training set, which is $\bigcup_{j,d} I(\text{NP}_{jd})$, where NP_{jd} is now the jth NP in training document d. NEG-SELECT is designed essentially to compute this set. Next, we examine the effects of this minimalist approach to negative sample selection.

Evaluation. We evaluate the coreference system with negative sample selection on the MUC-6 and MUC-7 coreference data sets in each case, training the coreference classifier on the 30 "dry run" texts, and applying the coreference resolution algorithm on the 20–30 "formal evaluation" texts. Results are shown in rows 1 and 2 of Table 2 where performance is reported in terms of recall, precision, and F-measure using the model-theoretic MUC scoring program (Vilain et al., 1995). The Baseline system employs no sample selection, i.e. all available training examples are used. Row 2 shows the performance of the Baseline after incorporating NEG-SELECT. With negative sample selection, the percentage of positive instances rises from 2% to 8% for the MUC-6 data set and from 2% to 7% for the MUC-7 data set. For both data sets, we see statistically significant increases in recall and statistically significant, but much larger drops in precision.[9] The resulting F-measure scores, however, increase non-trivially from 52.4 to 55.2 (for MUC-6), and from 41.3 to 46.0 (for MUC-7).[10]

4 Positive Sample Selection

Since not all of the coreference relationships derived from coreference chains are equally easy to identify, training a classifier using all possible coreference relationships can potentially lead to the induction of inaccurate rules. Given the observation that one antecedent is sufficient to resolve an anaphor, it may be desirable to learn only from easy positive instances. Similar observations are made by Harabagiu et al. (2001), who point out that intelligent selection of positive instances can potentially minimize the amount of knowledge required to perform coreference resolution accurately. They assume that the easiest types of coreference relationships to resolve are those that occur with high frequencies in the data. Consequently, they mine by hand three sets of coreference rules for covering positive instances from the training data by finding the coreference knowledge satisfied by the largest number of anaphor-antecedent pairs. While the Harabagiu et al. algorithm attempts to mine easy coreference rules from the data by hand, neither the rule creation process nor stopping conditions are precisely defined. In addition, a lot of human intervention is required to derive the rules. In this section, we describe an automatic positive sample selection algorithm that coarsely mimics the Harabagiu et al. algorithm by finding a *confident* antecedent for each anaphor. Overall, our goal is to avoid the inclusion of hard training instances by automating the process of deriving easy coreference rules from the data.

The Algorithm. The positive sample selection algorithm, POS-SELECT, is shown in Figure 2. It assumes the existence of a rule learner, L, that produces an ordered set of *positive* rules. POS-SELECT first uses L to induce a ruleset on the training instances and picks the first rule from the ruleset. For any training instance $i_{(NP_{id},NP_{jd})}$ correctly covered by this rule, an antecedent NP_{id} has been identified for the anaphor NP_{jd}. As a result, all (positive and negative) training instances formed with NP_{jd} as the anaphor are no longer needed and are subsequently removed from the training data.[11] The process is repeated until L cannot induce a rule to cover the remaining positive instances. The output of POS-SELECT is a set of positive rules selected during each iteration of the algorithm. Hence, positive sample selection in POS-SELECT is implicit in the sense that it is embedded within the rule induction process.

Algorithm POS-SELECT(L: positive rule learner,
                                        T: set of training instances)
FinalRuleSet := $\emptyset$;
AnaphorSet := $\emptyset$;
BestRule := NIL;
repeat
    BestRule := best rule among the ranked set
        of positive rules induced on T using L
    FinalRuleSet := FinalRuleSet $\cup$ BestRule
    // collect anaphors from instances that
    // are correctly covered by BestRule
    for $i_{(NP_{id},NP_{jd})} \in$ T **do**
        if $i_{(NP_{id},NP_{jd})}$ is covered by BestRule **and**
        class($i_{(NP_{id},NP_{jd})}$) = COREFERENT **then**
            AnaphorSet := AnaphorSet $\cup$ { NP_{jd}}
    // remove instances associated with the
    // anaphors covered by BestRule
    for $i_{(NP_{id},NP_{jd})} \in$ T **do**
        if $\text{NP}_{jd} \in$ AnaphorSet **then**
            T := T $\setminus$ { $i_{(NP_{id},NP_{jd})}$}
until L cannot induce any rule for the positives.
return FinalRuleSet

Figure 2: The POS-SELECT algorithm

Evaluation. Results are shown in rows 3 and 4 of Table 2. As in the previous experiments, the rule learner is RIPPER. We run the system twice, first

[9] Chi-square statistical significance tests are applied to changes in recall and precision throughout the paper. Unless otherwise noted, reported differences are at the 0.05 level or higher. The chi-square test is not applicable to F-measure.

[10] The F-measure score computed by the MUC scoring program is the harmonic mean of recall and precision.

[11] We speculate that retaining the negative instances would hurt performance, but this remains to be verified.

Experiments	Algorithms used	MUC-6			MUC-7		
		R	P	F	R	P	F
Baseline	—	40.7	73.5	52.4	27.2	86.3	41.3
Neg-Only	NEG-SELECT	46.5	67.8	55.2	37.4	59.7	46.0
Pos-Only	POS-SELECT	53.1	80.8	64.1	41.1	78.0	53.8
Combined	NEG-SELECT+POS-SELECT	63.4	76.3	69.3	59.5	55.1	57.2
Pruning	NEG-SELECT+POS-SELECT+RULE-SELECT	63.3	76.9	69.5	54.2	76.3	63.4
More Training	NEG-SELECT+POS-SELECT	64.8	70.6	67.6	60.0	55.7	57.8

Table 2: Effects of sample selection and error-driven pruning.

with POS-SELECT only and then with both POS-SELECT and NEG-SELECT. With POS-SELECT only, the system achieves an F-measure of 64.1 (for MUC-6) and 53.8 (for MUC-7). When POS-SELECT and NEG-SELECT are used in combination, however, the system achieves an F-measure of 69.3 (for MUC-6) and 57.2 (for MUC-7).

Discussion. The experimental results are largely consistent with our hypothesis. System performance improves dramatically with positive sample selection using POS-SELECT both in the absence and presence of negative sample selection. Without negative sample selection, F-measure increases from 52.4 to 64.1 (for MUC-6), and from 41.3 to 53.8 (for MUC-7). Similarly, with negative sample selection, F-measure increases from 55.2 to 69.3 (for MUC-6), and from 46.0 to 57.2 (for MUC-7). In addition, our results indicate that applying both negative and positive sample selection leads to better performance than applying positive sample selection alone: F-measure increases from 64.1 to 69.3, and from 53.8 to 57.2 for the MUC-6 and MUC-7 data sets, respectively. Nevertheless, reducing the number of negative instances (via negative sample selection) improves recall but damages precision: we see statistically significant gains in recall and statistically significant drops in precision for both data sets. In particular, precision drops precipitously from 78.0 to 55.1 for the MUC-7 data set. We hypothesize that POS-SELECT does not guarantee that hard positive instances will be avoided and that the inclusion of these hard instances is responsible for the poorer precision of the system. Anaphors that do not have easy antecedents can never be removed automatically via the induction of new rules using POS-SELECT. In fact, RIPPER will possibly induce rules to handle these hard instances as long as such kind of anaphors occur sufficiently frequently in the data set relative to the number of negative instances.[12] Although it might be beneficial to acquire these rules at the classification level (according to the learning algorithm), they can be detrimental to system performance at the clustering level, especially if the rules cover a large number of examples with a lot of exceptions. Consequently, it is necessary to know which rules are worthy of keeping at the clustering level and not the classification level. We will address this issue in the next section.

5 Pruning the Coreference Ruleset

As noted in the introduction, machine learning approaches to coreference resolution that rely only on pairwise NP coreference classifiers will not necessarily enforce the transitivity constraint inherent in the coreference relation. Although approaches to coreference resolution that rely only on clustering could easily enforce transitivity (as in Cardie and Wagstaff (1999)), they have not performed as well as state-of-the-art approaches to coreference. In this section, we propose a method for resolving this conflict: we introduce an error-driven rule pruning algorithm that considers rules induced by the coreference classifier and discards those that cause the ruleset to perform poorly with respect to the global, clustering-level coreference scoring function.

The Algorithm. The error-driven pruning algorithm is inspired by the backward elimination algorithm commonly used for feature selection (see Blum and Langley (1997)) and is shown in Figure 3. The algorithm, RULE-SELECT, takes as input a ruleset learned from a training corpus for performing coreference resolution, a pruning corpus (disjoint from the training corpus), and a clustering-level

[12] More precisely, RIPPER will induce a new rule if the rule is more than 50% accurate and the resulting description length is fewer than 64 bits larger than the smallest description length obtained so far.

```
Algorithm RULE-SELECT(R: ruleset,
                      P: pruning corpus,
                      S: scoring function)
BestScore := score of the coreference system
             using R on P w.r.t. S;
r := NIL;
repeat
   r := the rule in R whose removal yields a
      ruleset with which the coreference system
      achieves the best score b on P w.r.t. S.
   if b > BestScore then
      BestScore := b;
      R := R \ {r}
   else break
while true
return R
```

Figure 3: The RULE-SELECT algorithm

coreference scoring function that is the same as the one being used for evaluating the final output of the system.[13] At each iteration, RULE-SELECT greedily discards the rule whose removal yields a ruleset with which the coreference system performs the best (with respect to the coreference scoring function) on the pruning corpus. As a hill-climbing procedure, the algorithm terminates when removal of any of the rules in the ruleset fails to improve performance. In contrast to most existing algorithms for coreference resolution, RULE-SELECT establishes a tighter connection between the classification- and clustering-level decisions for coreference resolution and ensures that system performance is optimized with respect to the coreference scoring function. We hypothesize that this optimization of the coreference classifier will improve performance of the resulting coreference system, in particular by increasing its precision.

Evaluation and Discussion. Results are shown in row 5 of Table 2. In the Pruning experiment, the MUC-7 formal evaluation corpus is the pruning corpus for the MUC-6 run; the MUC-6 formal evaluation corpus is the pruning corpus for the MUC-7 run. In addition, the quantity that RULE-SELECT optimizes for a given ruleset is the F-measure returned by the MUC scoring function.[14] In comparison to the Combined results, we see an improvement of 0.2% (for MUC-6) and 6.2% (for MUC-7) in F-measure. In particular, we see statistically significant gains in precision (from 55.1 to 73.6) and statistically significant, but much smaller, drops in recall (from 59.5 to 54.2) for the MUC-7 data set. In general, our results support the hypothesis that rule pruning can be used to improve system performance; moreover, the technique is especially effective at enhancing the precision of the system. However, performance gains may be negligible when pruning is used in systems with high precision, as can be seen from the results for the MUC-6 data set.

To determine whether performance improvements are instead attributable to the availability of additional "training" data provided by the pruning corpus, we train a classifier (using the same setting as the Combined experiments) on both the training and the pruning corpora. The performance of the system using this unpruned ruleset is shown in the last row of Table 2. In comparison to the Combined results, F-measure drops from 69.3 to 67.6 (for MUC-6), and rises from 57.2 and 57.8 (for MUC-7). These results indicate that the RULE-SELECT algorithm has made a more effective use of the additional data than the learning algorithm without rule pruning by exploiting the feedback provided by the scoring function.

6 Conclusions

We have examined three problems with recasting noun phrase coreference resolution as a classification task. To handle these problems, we presented a minimalist negative sample selection algorithm to reduce the skewness of the class distributions, and an automatic positive sample selection algorithm to select easy positive instances. In addition, our experiments indicate that the positive sample selection algorithm does not guarantee that hard instances can be entirely excluded. As a result, we proposed an error-driven rule pruning algorithm that can effectively enhance the precision of the system by dis-

[13]Importantly, RULE-SELECT assumes *no* knowledge of the inner workings of the scoring function.

[14]RULE-SELECT can be used in conjunction with *any* coreference scoring function. The MUC scorer is chosen here to facilitate comparison with previous results.

carding rules that cause the ruleset to perform poorly with respect to the coreference scoring function. The resulting system outperformed the best MUC-6 and MUC-7 coreference systems as well as the best-performing learning-based system on the corresponding MUC data sets. Nevertheless, there is substantial room for improvement. For example, it is important to know how sensitive system performance is with respect to the size of the pruning corpus. In addition, although we use RIPPER as the underlying learning algorithm in our coreference system, we expect that the techniques described in this paper can be used in conjunction with other learning algorithms. We plan to explore this possibility in future work.

Acknowledgments

We thank two anonymous reviewers for their helpful comments and Hwee Tou Ng for explaining the method of training instance selection employed by his coreference system. This work was supported in part by DARPA TIDES contract N66001-00-C-8009, and NSF Grants 0081334 and 0074896.

References

C. Aone and S. W. Bennett. 1995. Evaluating Automated and Manual Acquisition of Anaphora Resolution Strategies. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 122–129.

A. Blum and P. Langley. 1997. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, pages 245–271.

C. Cardie and N. Howe. 1997. Improving minority class prediction using case-specific feature weights. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 57–65.

C. Cardie and K. Wagstaff. 1999. Noun Phrase Coreference as Clustering. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*, pages 82–89.

W. Cohen. 1995. Fast Effective Rule Induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, San Francisco, CA.

A. Dempster. 1968. A generalization of Bayesian inference. *Journal of the Royal Statistical Society*, 30:205–247.

T. Fawcett. 1996. *Learning with skewed class distributions — summary of responses*. Machine Learning List: Vol. 8, No. 20.

D. F. Gordon and D. Perlis. 1989. Explicitly biased generalization. *Computational Intelligence*, 5:67–81.

S. Harabagiu, R. Bunescu, and S. Maiorano. 2001. Text and Knowledge Mining for Coreference Resolution. In *Proceedings of the Second Meeting of the North America Chapter of the Association for Computational Linguistics (NAACL-2001)*, pages 55–62.

A. Kehler. 1997. Probabilistic Coreference in Information Extraction. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 163–173.

M. Kubat and S. Matwin. 1997. Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In *Proceedings of the 14th International Conference on Machine Learning (ICML-97)*, pages 179–186.

J. McCarthy and W. Lehnert. 1995. Using Decision Trees for Coreference Resolution. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence*, pages 1050–1055.

MUC-6. 1995. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kaufmann, San Francisco, CA.

MUC-7. 1998. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Morgan Kaufmann, San Francisco, CA.

V. Ng and C. Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

M. Pazzani, C. Merz, P. Murphy, K. Ali, T. Hume, and C. Brunk. 1994. Reducing Misclassification Costs. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 217–225.

W. M. Soon, H. T. Ng, and D. C. Y. Lim. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4):521–544.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 45–52. Morgan Kaufmann.