Proceedings of the 40th Annual Meeting of the Association for
Computational Linguistics (ACL), Philadelphia, PA, July 2002, pp. 104-111.

Improving Machine Learning Approaches to Coreference Resolution

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

Abstract

We present a noun phrase coreference system that extends the work of Soon et al. (2001) and, to our knowledge, produces the best results to date on the MUC-6 and MUC-7 coreference resolution data sets — F-measures of 70.4 and 63.4, respectively. Improvements arise from two sources: extra-linguistic changes to the learning framework and a large-scale expansion of the feature set to include more sophisticated linguistic knowledge.

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)). Specifically, a pair of NPs is classified as co-referring or not based on constraints that are learned from an annotated corpus. A separate clustering mechanism then coordinates the possibly contradictory pairwise classifications and constructs a partition on the set of NPs. Soon et al. (2001), for example, apply an NP coreference system based on decision tree induction to two standard coreference resolution data sets (MUC-6, 1995; MUC-7, 1998), achieving performance comparable to the best-performing knowledge-based coreference engines. Perhaps surprisingly, this was accomplished in a decidedly knowledge-lean manner — the learning algorithm has access to just 12 surface-level features.

This paper presents an NP coreference system that investigates two types of extensions to the Soon et al. corpus-based approach. First, we propose and evaluate three extra-linguistic modifications to the machine learning framework, which together provide substantial and statistically significant gains in coreference resolution precision. Second, in an attempt to understand whether incorporating additional knowledge can improve the performance of a corpus-based coreference resolution system, we expand the Soon et al. feature set from 12 features to an arguably deeper set of 53. We propose additional lexical, semantic, and knowledge-based features; most notably, however, we propose 26 additional grammatical features that include a variety of linguistic constraints and preferences. Although the use of similar knowledge sources has been explored in the context of both pronoun resolution (e.g. Lappin and Leass (1994)) and NP coreference resolution (e.g. Grishman (1995), Lin (1995)), most previous work treats linguistic constraints as broadly and unconditionally applicable hard constraints. Because sources of linguistic information in a learning-based system are represented as features, we can, in contrast, incorporate them selectively rather than as universal hard constraints.

Our results using an expanded feature set are mixed. First, we find that performance drops significantly when using the full feature set, even though the learning algorithms investigated have built-in feature selection mechanisms. We demonstrate empirically that the degradation in performance can be attributed, at least in part, to poor performance on common noun resolution. A manually selected subset of 22–26 features, however, is shown to provide significant gains in performance when chosen specifically to improve precision on common noun resolution. Overall, the learning framework and linguistic knowledge source modifications boost performance of Soon's learning-based coreference resolution approach from an F-measure of 62.6 to 70.4, and from 60.4 to 63.4 for the MUC-6 and MUC-7 data sets, respectively. To our knowledge, these are the best results reported to date on these data sets for the full NP coreference problem.[1]

The rest of the paper is organized as follows. In sections 2 and 3, we present the baseline coreference system and explore extra-linguistic modifications to the machine learning framework. Section 4 describes and evaluates the expanded feature set. We conclude with related and future work in Section 5.

2 The Baseline Coreference System

Our baseline coreference system attempts to duplicate both the approach and the knowledge sources employed in Soon et al. (2001). More specifically, it employs the standard combination of classification and clustering described above.

Building an NP coreference classifier. We use the C4.5 decision tree induction system (Quinlan, 1993) to train a classifier that, given a description of two NPs in a document, NP_i and NP_j, decides whether or not they are coreferent. Each training instance represents the two NPs under consideration and consists of the 12 Soon et al. features, which are described in Table 1. Linguistically, the features can be divided into four groups: lexical, grammatical, semantic, and positional.[2] 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. We follow the procedure employed in Soon et al. to create the training data: we rely on coreference chains from the MUC answer keys to create (1) a *positive instance* for each anaphoric noun phrase, NP_j, and its closest preceding antecedent, NP_i; and (2) a *negative instance* for NP_j paired with each of the intervening NPs, NP_{i+1}, NP_{i+2},. . ., NP_{j-1}. This method of negative instance selection is further described in Soon et al. (2001); it is designed to operate in conjunction with their method for creating coreference chains, which is explained next.

Applying the classifier to create coreference chains. After training, the decision tree is used by a clustering algorithm to impose a partitioning on all NPs in the test texts, creating one cluster for each set of coreferent NPs. As in Soon et al., texts are processed from left to right. Each NP encountered, NP_j, is compared in turn to each preceding NP, NP_i, from right to left. For each pair, a test instance is created as during training and is presented to the coreference classifier, which returns a number between 0 and 1 that indicates the likelihood that the two NPs are coreferent.[3] NP pairs with class values above 0.5 are considered COREFERENT; otherwise the pair is considered NOT COREFERENT. The process terminates as soon as an antecedent is found for NP_j or the beginning of the text is reached.

2.1 Baseline Experiments

We evaluate the Duplicated Soon Baseline system using the standard MUC-6 (1995) and MUC-7 (1998) coreference corpora, training the coreference classifier on the 30 "dry run" texts, and applying the coreference resolution algorithm on the 20–30 "formal evaluation" texts. The MUC-6 corpus produces a training set of 26455 instances (5.4% positive) from 4381 NPs and a test set of 28443 instances (5.2% positive) from 4565 NPs. For the MUC-7 corpus, we obtain a training set of 35895 instances (4.4% positive) from 5270 NPs and a test set of 22699 instances (3.9% positive) from 3558 NPs.

Results are shown in Table 2 (Duplicated Soon Baseline) where performance is reported in terms of recall, precision, and F-measure using the model-theoretic MUC scoring program (Vilain et al., 1995).

[1] Results presented in Harabagiu et al. (2001) are higher than those reported here, but assume that all and only the noun phrases involved in coreference relationships are provided for analysis by the coreference resolution system. We presume no preprocessing of the training and test documents.

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

[3] We convert the binary class value using the smoothed ratio $\frac{p+1}{t+2}$, where p is the number of positive instances and t is the total number of instances contained in the corresponding leaf node.

Feature Type	Feature	Description
Lexical	SOON_STR	C if, after discarding determiners, the string denoting NP_i matches that of NP_j; else I.
Grammatical	PRONOUN_1*	Y if NP_i is a pronoun; else N.
	PRONOUN_2*	Y if NP_j is a pronoun; else N.
	DEFINITE_2	Y if NP_j starts with the word "the;" else N.
	DEMONSTRATIVE_2	Y if NP_j starts with a demonstrative such as "this," "that," "these," or "those;" else N.
	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.
	BOTH_PROPER_NOUNS*	C if both NPs are proper names; NA if exactly one NP is a proper name; else I.
	APPOSITIVE*	C if the NPs are in an appositive relationship; else I.
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.

Table 1: Feature Set for the Duplicated Soon Baseline 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. *'d features are in the hand-selected feature set (see Section 4) for at least one classifier/data set combination.

The system achieves an F-measure of 66.3 and 61.2 on the MUC-6 and MUC-7 data sets, respectively. Similar, but slightly worse performance was obtained using RIPPER (Cohen, 1995), an information-gain-based rule learning system. Both sets of results are at least as strong as the original Soon results (row one of Table 2), indicating indirectly that our Baseline system is a reasonable duplication of that system.[4] In addition, the trees produced by Soon and by our Duplicated Soon Baseline are essentially the same, differing only in two places where the Baseline system imposes additional conditions on coreference.

The primary reason for improvements over the original Soon system for the MUC-6 data set appears to be our higher upper bound on recall (93.8% vs. 89.9%), due to better identification of NPs. For MUC-7, our improvement stems from increases in precision, presumably due to more accurate feature value computation.

[4]In all of the experiments described in this paper, default settings for all C4.5 parameters are used. Similarly, all RIPPER parameters are set to their default value except that classification rules are induced for both the positive and negative instances.

3 Modifications to the Machine Learning Framework

This section studies the effect of three changes to the general machine learning framework employed by Soon et al. with the goal of improving precision in the resulting coreference resolution systems.

Best-first clustering. Rather than a right-to-left search from each anaphoric NP for the first coreferent NP, we hypothesized that a right-to-left search for a *highly likely antecedent* might offer more precise, if not generally better coreference chains. As a result, we modify the coreference clustering algorithm to select as the antecedent of NP_j the NP with the highest coreference likelihood value from among preceding NPs with coreference class values above 0.5.

Training set creation. For the proposed best-first clustering to be successful, however, a different method for training instance selection would be needed: rather than generate a positive training example for each anaphoric NP and its **closest** antecedent, we instead generate a positive training example for its **most confident** antecedent. More specifically, for a non-pronominal NP, we assume that the most confident antecedent is the closest **non-**

	C4.5						RIPPER					
	MUC-6			MUC-7			MUC-6			MUC-7		
System Variation	R	P	F	R	P	F	R	P	F	R	P	F
Original Soon et al.	58.6	67.3	62.6	56.1	65.5	60.4	-	-	-	-	-	-
Duplicated Soon Baseline	62.4	70.7	66.3	55.2	68.5	61.2	60.8	68.4	64.3	54.0	69.5	60.8
Learning Framework	62.4	73.5	67.5	56.3	71.5	63.0	60.8	75.3	67.2	55.3	73.8	**63.2**
String Match	60.4	74.4	66.7	54.3	72.1	62.0	58.5	74.9	65.7	48.9	73.2	58.6
Training Instance Selection	61.9	70.3	65.8	55.2	68.3	61.1	61.3	70.4	65.5	54.2	68.8	60.6
Clustering	62.4	70.8	66.3	56.5	69.6	62.3	60.5	68.4	64.2	55.6	70.7	62.2
All Features	70.3	58.3	63.8	65.5	58.2	61.6	67.0	62.2	64.5	61.9	60.6	61.2
Pronouns only	–	66.3	–	–	62.1	–	–	71.3	–	–	62.0	–
Proper Nouns only	–	84.2	–	–	77.7	–	–	85.5	–	–	75.9	–
Common Nouns only	–	40.1	–	–	45.2	–	–	43.7	–	–	48.0	–
Hand-selected Features	64.1	74.9	**69.1**	57.4	70.8	**63.4**	64.2	78.0	**70.4**	55.7	72.8	63.1
Pronouns only	–	67.4	–	–	54.4	–	–	77.0	–	–	60.8	–
Proper Nouns only	–	93.3	–	–	86.6	–	–	95.2	–	–	88.7	–
Common Nouns only	–	63.0	–	–	64.8	–	–	62.8	–	–	63.5	–

Table 2: Results for the MUC-6 and MUC-7 data sets using C4.5 and RIPPER. **R**ecall, **P**recision, and **F**-measure are provided. Results in boldface indicate the best results obtained for a particular data set and classifier combination.

pronominal preceding antecedent. For pronouns, we assume that the most confident antecedent is simply its closest preceding antecedent. Negative examples are generated as in the Baseline system.[5]

String match feature. Soon's string match feature (SOON_STR) tests whether the two NPs under consideration are the same string after removing determiners from each. We hypothesized, however, that splitting this feature into several primitive features, depending on the type of NP, might give the learning algorithm additional flexibility in creating coreference rules. Exact string match is likely to be a better coreference predictor for proper names than it is for pronouns, for example. Specifically, we replace the SOON_STR feature with three features — PRO_STR, PN_STR, and WORDS_STR — which restrict the application of string matching to pronouns, proper names, and non-pronominal NPs, respectively. (See the first entries in Table 3.) Although similar feature splits might have been considered for other features (e.g. GENDER and NUMBER), only the string match feature was tested here.

Results and discussion. Results on the learning framework modifications are shown in Table 2 (third block of results). When used in combination, the modifications consistently provide statistically significant gains in precision over the Baseline system without any loss in recall.[6] As a result, we observe reasonable increases in F-measure for both classifiers and both data sets. When using RIPPER, for example, performance increases from 64.3 to 67.2 for the MUC-6 data set and from 60.8 to 63.2 for MUC-7. Similar, but weaker, effects occur when applying each of the learning framework modifications to the Baseline system in isolation. (See the indented Learning Framework results in Table 2.)

Our results provide direct evidence for the claim (Mitkov, 1997) that the extra-linguistic strategies employed to combine the available linguistic knowledge sources play an important role in computational approaches to coreference resolution. In particular, our results suggest that additional performance gains might be obtained by further investigating the interaction between training instance selection, feature selection, and the coreference clustering algorithm.

4 NP Coreference Using Many Features

This section describes the second major extension to the Soon approach investigated here: we explore the effect of including 41 additional, potentially useful knowledge sources for the coreference resolution classifier (Table 3). The features were not derived empirically from the corpus, but were based on common-sense knowledge and linguistic intuitions

[5]This new method of training set creation slightly alters the class value distribution in the training data: for the MUC-6 corpus, there are now 27654 training instances of which 5.2% are positive; for the MUC-7 corpus, there are now 37870 training instances of which 4.2% are positive.

[6]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.

regarding coreference. Specifically, we increase the number of lexical features to nine to allow more complex NP string matching operations. In addition, we include four new semantic features to allow finer-grained semantic compatibility tests. We test for ancestor-descendent relationships in WordNet (SUBCLASS), for example, and also measure the WordNet graph-traversal distance (WNDIST) between NP_j and NP_i. Furthermore, we add a new positional feature that measures the distance in terms of the number of paragraphs (PARANUM) between the two NPs.

The most substantial changes to the feature set, however, occur for grammatical features: we add 26 new features to allow the acquisition of more sophisticated syntactic coreference resolution rules. Four features simply determine NP type, e.g. are both NPs definite, or pronouns, or part of a quoted string? These features allow other tests to be conditioned on the types of NPs being compared. Similarly, three new features determine the grammatical role of one or both of the NPs. Currently, only tests for clausal subjects are made. Next, eight features encode traditional linguistic (hard) constraints on coreference. For example, coreferent NPs must agree both in gender and number (AGREEMENT); cannot SPAN one another (e.g. "government" and "government officials"); and cannot violate the BINDING constraints. Still other grammatical features encode general linguistic preferences either for or against coreference. For example, an indefinite NP (that is not in apposition to an anaphoric NP) is not likely to be coreferent with any NP that precedes it (ARTICLE). The last subset of grammatical features encodes slightly more complex, but generally non-linguistic heuristics. For instance, the CONTAINS_PN feature effectively disallows coreference between NPs that contain distinct proper names but are not themselves proper names (e.g. "IBM executives" and "Microsoft executives").

Two final features make use of an in-house naive pronoun resolution algorithm (PRO_RESOLVE) and a rule-based coreference resolution system (RULE_RESOLVE), each of which relies on the original and expanded feature sets described above.

Results and discussion. Results using the expanded feature set are shown in the All Features block of Table 2. These and all subsequent results also incorporate the learning framework changes from Section 3. In comparison, we see statistically significant increases in recall, but much larger decreases in precision. As a result, F-measure drops precipitously for both learning algorithms and both data sets. A closer examination of the results indicates very poor precision on common nouns in comparison to that of pronouns and proper nouns. (See the indented All Features results in Table 2.[7]) In particular, the classifiers acquire a number of low-precision rules for common noun resolution, presumably because the current feature set is insufficient. For instance, a rule induced by RIPPER classifies two NPs as coreferent if the first NP is a proper name, the second NP is a definite NP in the subject position, and the two NPs have the same semantic class and are at most one sentence apart from each other. This rule covers 38 examples, but has 18 exceptions. In comparison, the Baseline system obtains much better precision on common nouns (i.e. 53.3 for MUC-6/RIPPER and 61.0 for MUC-7/RIPPER with lower recall in both cases) where the primary mechanism employed by the classifiers for common noun resolution is its high-precision string matching facility. Our results also suggest that data fragmentation is likely to have contributed to the drop in performance (i.e. we increased the number of features without increasing the size of the training set). For example, the decision tree induced from the MUC-6 data set using the Soon feature set (Learning Framework results) has 16 leaves, each of which contains 1728 instances on average; the tree induced from the same data set using all of the 53 features, on the other hand, has 86 leaves with an average of 322 instances per leaf.

Hand-selected feature sets. As a result, we next evaluate a version of the system that employs manual feature selection: for each classifier/data set combination, we discard features used primarily to induce low-precision rules for common noun resolution and re-train the coreference classifier using the reduced feature set. Here, feature selection does not depend on a separate development corpus and

[7]For each of the NP-type-specific runs, we measure overall coreference performance, but restrict NP_j to be of the specified type. As a result, recall and F-measure for these runs are not particularly informative.

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.
		WORDS_STR	C if both NPs are non-pronominal and are the same string; else I.
		SOON_STR_NONPRO*	C if both NPs are non-pronominal and the string of NP$_i$ matches that of NP$_j$; else I.
		WORD_OVERLAP	C if the intersection between the content words in NP$_i$ and NP$_j$ is not empty; else I.
		MODIFIER	C if the prenominal modifiers of one NP are a subset of the prenominal modifiers of the other; else I.
		PN_SUBSTR	C if both NPs are proper names and one NP is a proper substring (w.r.t. content words only) of the other; else I.
		WORDS_SUBSTR	C if both NPs are non-pronominal and one NP is a proper substring (w.r.t. content words only) of the other; else I.
Grammatical	NP type	BOTH_DEFINITES	C if both NPs start with "the;" I if neither start with "the;" else NA.
		BOTH_EMBEDDED	C if both NPs are prenominal modifiers ; I if neither are prenominal modifiers; else NA.
		BOTH_IN_QUOTES	C if both NPs are part of a quoted string; I if neither are part of a quoted string; else NA.
		BOTH_PRONOUNS*	C if both NPs are pronouns; I if neither are pronouns, else NA.
	role	BOTH_SUBJECTS	C if both NPs are grammatical subjects; I if neither are subjects; else NA.
		SUBJECT_1*	Y if NP$_i$ is a subject; else N.
		SUBJECT_2	Y if NP$_j$ is a subject; else N.
	linguistic constraints	AGREEMENT*	C if the NPs agree in both gender and number; I if they disagree in both gender and number; else NA.
		ANIMACY*	C if the NPs match in animacy; else I.
		MAXIMALNP*	I if both NPs have the same maximal NP projection; else C.
		PREDNOM*	C if the NPs form a predicate nominal construction; else I.
		SPAN*	I if one NP spans the other; else C.
		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.
		SYNTAX*	I if the NPs have incompatible values for the BINDING, CONTRAINDICES, SPAN or MAXIMALNP constraints; else C.
	ling. prefs	INDEFINITE*	I if NP$_j$ is an indefinite and not appositive; else C.
		PRONOUN	I if NP$_i$ is a pronoun and NP$_j$ is not; else C.
	heuristics	CONSTRAINTS*	C if the NPs agree in GENDER and NUMBER and do not have incompatible values for CONTRAINDICES, SPAN, ANIMACY, PRONOUN, and CONTAINS_PN; I if the NPs have incompatible values for any of the above features; else NA.
		CONTAINS_PN	I if both NPs are not proper names but contain proper names that mismatch on every word; else C.
		DEFINITE_1	Y if NP$_i$ starts with "the;" else N.
		EMBEDDED_1*	Y if NP$_i$ is an embedded noun; else N.
		EMBEDDED_2	Y if NP$_j$ is an embedded noun; else N.
		IN_QUOTE_1	Y if NP$_i$ is part of a quoted string; else N.
		IN_QUOTE_2	Y if NP$_j$ is part of a quoted string; else N.
		PROPER_NOUN	I if both NPs are proper names, but mismatch on every word; else C.
		TITLE*	I if one or both of the NPs is a title; else C.
Semantic		CLOSEST_COMP	C if NP$_i$ is the closest NP preceding NP$_j$ that has the same semantic class as NP$_j$ and the two NPs do not violate any of the linguistic constraints; else I.
		SUBCLASS	C if the NPs have different head nouns but have an ancestor-descendent relationship in WordNet; else I.
		WNDIST	Distance between NP$_i$ and NP$_j$ in WordNet (using the first sense only) when they have an ancestor-descendent relationship but have different heads; else infinity.
		WNSENSE	Sense number in WordNet for which there exists an ancestor-descendent relationship between the two NPs when they have different heads; else infinity.
Pos		PARANUM	Distance between the NPs in terms of the number of paragraphs.
Other		PRO_RESOLVE*	C if NP$_j$ is a pronoun and NP$_i$ is its antecedent according to a naive pronoun resolution algorithm; else I.
		RULE_RESOLVE	C if the NPs are coreferent according to a rule-based coreference resolution algorithm; else I.

Table 3: Additional features for NP coreference. As before, *'d features are in the hand-selected feature set for at least one classifier/data set combination.

is guided solely by inspection of the features associated with low-precision rules induced from the training data. In current work, we are automating this feature selection process, which currently employs a fair amount of user discretion, e.g. to determine a precision cut-off. Features in the hand-selected set for at least one of the tested system variations are *'d in Tables 1 and 3.

In general, we hypothesized that the hand-selected features would reclaim precision, hopefully without losing recall. For the most part, the experimental results support this hypothesis. (See the Hand-selected Features block in Table 2.) In comparison to the All Features version, we see statistically significant gains in precision and statistically significant, but much smaller, drops in recall, producing systems with better F-measure scores. In addition, precision on common nouns rises substantially, as expected. Unfortunately, the hand-selected features precipitate a large drop in precision for pronoun resolution for the MUC-7/C4.5 data set. Additional analysis is required to determine the reason for this.

Moreover, the Hand-selected Features produce the highest scores posted to date for both the MUC-6 and MUC-7 data sets: F-measure increases w.r.t. the Baseline system from 64.3 to 70.4 for MUC-6/RIPPER, and from 61.2 to 63.4 for MUC-7/C4.5. In one variation (MUC-7/RIPPER), however, the Hand-selected Features slightly underperforms the Learning Framework modifications (F-measure of 63.1 vs. 63.2) although changes in recall and precision are not statistically significant. Overall, our results indicate that pronoun and especially common noun resolution remain important challenges for coreference resolution systems. Somewhat disappointingly, only four of the new grammatical features corresponding to linguistic constraints and preferences are selected by the symbolic learning algorithms investigated: AGREEMENT, ANIMACY, BINDING, and MAXIMALNP.

Discussion. In an attempt to gain additional insight into the difference in performance between our system and the original Soon system, we compare the decision tree induced by each for the MUC-6 data set.[8] For our system, we use the tree induced on the hand-selected features (Figure 1). The two trees are fairly different. In particular, our tree makes use of many of the features that are not present in the original Soon feature set. The root feature for Soon, for example, is the general string match feature (SOON_STR); splitting the SOON_STR feature into three primitive features promotes the ALIAS feature to the root of our tree, on the other hand. In addition, given two non-pronominal, matching NPs (SOON_STR_NONPRO=C), our tree requires an additional test on ANIMACY before considering the two NPs coreferent; the Soon tree instead determines two NPs to be coreferent as long as they are the same string. Pronoun resolution is also performed quite differently by the two trees, although both consider two pronouns coreferent when their strings match. Finally, intersentential and intrasentential pronominal references are possible in our system while intersentential pronominal references are largely prohibited by the Soon system.

```
ALIAS = C: + (347.0/23.8)
ALIAS = I:
|  SOON_STR_NONPRO = C:
|  |  ANIMACY = NA: - (4.0/2.2)
|  |  ANIMACY = I: + (0.0)
|  |  ANIMACY = C: + (259.0/45.8)
|  SOON_STR_NONPRO = I:
|  |  PRO_STR = C: + (39.0/2.6)
|  |  PRO_STR = I:
|  |  |  PRO_RESOLVE = C:
|  |  |  |  EMBEDDED_1 = Y: - (7.0/3.4)
|  |  |  |  EMBEDDED_1 = N:
|  |  |  |  |  PRONOUN_1 = Y:
|  |  |  |  |  |  ANIMACY = NA: - (6.0/2.3)
|  |  |  |  |  |  ANIMACY = I: - (1.0/0.8)
|  |  |  |  |  |  ANIMACY = C: + (10.0/3.5)
|  |  |  |  |  PRONOUN_1 = N:
|  |  |  |  |  |  MAXIMALNP = C: + (108.0/18.2)
|  |  |  |  |  |  MAXIMALNP = I:
|  |  |  |  |  |  |  WNCLASS = NA: - (5.0/1.2)
|  |  |  |  |  |  |  WNCLASS = I: + (0.0)
|  |  |  |  |  |  |  WNCLASS = C: + (12.0/3.6)
|  |  |  PRO_RESOLVE = I:
|  |  |  |  APPOSITIVE = I: - (26806.0/713.8)
|  |  |  |  APPOSITIVE = C:
|  |  |  |  |  GENDER = NA: + (28.0/2.6)
|  |  |  |  |  GENDER = I: + (5.0/3.2)
|  |  |  |  |  GENDER = C: - (17.0/3.7)
```

Figure 1: Decision Tree using the Hand-selected feature set on the MUC-6 data set.

5 Conclusions

We investigate two methods to improve existing machine learning approaches to the problem of

[8]Soon et al. (2001) present only the tree learned for the MUC-6 data set.

noun phrase coreference resolution. First, we propose three extra-linguistic modifications to the machine learning framework, which together consistently produce statistically significant gains in precision and corresponding increases in F-measure. Our results indicate that coreference resolution systems can improve by effectively exploiting the interaction between the classification algorithm, training instance selection, and the clustering algorithm. We plan to continue investigations along these lines, developing, for example, a true best-first clustering coreference framework and exploring a "supervised clustering" approach to the problem. In addition, we provide the learning algorithms with many additional linguistic knowledge sources for coreference resolution. Unfortunately, we find that performance drops significantly when using the full feature set; we attribute this, at least in part, to the system's poor performance on common noun resolution and to data fragmentation problems that arise with the larger feature set. Manual feature selection, with an eye toward eliminating low-precision rules for common noun resolution, is shown to reliably improve performance over the full feature set and produces the best results to date on the MUC-6 and MUC-7 coreference data sets — F-measures of 70.4 and 63.4, respectively. Nevertheless, there is substantial room for improvement. As noted above, for example, it is important to automate the precision-oriented feature selection procedure as well as to investigate other methods for feature selection. We also plan to investigate previous work on common noun phrase interpretation (e.g. Sidner (1979), Harabagiu et al. (2001)) as a means of improving common noun phrase resolution, which remains a challenge for state-of-the-art coreference resolution systems.

Acknowledgments

Thanks to three anonymous reviewers for their comments and, in particular, for suggesting that we investigate data fragmentation issues. 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.

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

R. Grishman. 1995. The NYU System for MUC-6 or Where's the Syntax? In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*.

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.

S. Lappin and H. Leass. 1994. An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*, 20(4):535–562.

D. Lin. 1995. University of Manitoba: Description of the PIE System as Used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*.

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.

R. Mitkov. 1997. Factors in anaphora resolution: they are not the only things that matter. A case study based on two different approaches. In *Proceedings of the ACL'97/EACL'97 Workshop on Operational Factors in Practical, Robust Anaphora Resolution*.

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.

J. R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

C. Sidner. 1979. *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse*. PhD Thesis, Massachusetts Institute of Technology.

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, San Francisco, CA. Morgan Kaufmann.