Proceedings of the 19th International Conference on
Computational Linguistics (COLING), Taipei, Taiwan, August 2002, pp. 730-736.

Identifying Anaphoric and Non-Anaphoric Noun Phrases to Improve 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 supervised learning approach to identification of anaphoric and non-anaphoric noun phrases and show how such information can be incorporated into a coreference resolution system. The resulting system outperforms the best MUC-6 and MUC-7 coreference resolution systems on the corresponding MUC coreference data sets, obtaining F-measures of 66.2 and 64.0, respectively.

1 Introduction

Noun phrase coreference resolution refers to the problem of determining which noun phrases refer to each real-world entity mentioned in a document. While there appears to be consensus among researchers that the difficulty of this problem lies in its dependence on sophisticated world knowledge, recent work has shown that coreference resolution can be performed with high precision subject to at least one strong assumption — that the coreference resolution system knows *a priori* all and only the noun phrases (NPs) involved in coreference relationships for the document under analysis (Harabagiu et al., 2001). Although current machine learning approaches to coreference resolution have performed quite well (e.g. Aone and Bennett (1995), McCarthy and Lehnert (1995), Soon et al. (2001)), none explicitly addresses this assumption. Nonetheless, several researchers have tackled the closely related task of classifying whether or not a given discourse entity is anaphoric[1] (e.g. Paice and Husk (1987), Lappin and Leass (1994), Kennedy and Boguraev (1996), Denber (1998), Bean and Riloff (1999), Vieira and Poesio (2000), Evans (2001)). For the sake of brevity, we will henceforth refer to this task as anaphoricity determination.

The purpose of this paper is two-fold. First, we present a new learning-based method for anaphoricity determination. Our approach has a number of advantages over existing approaches to the problem:

- *The approach is supervised, so no hand-coded heuristics are required.* In contrast, the anaphoricity determination algorithms proposed by Paice and Husk (1987), Lappin and Leass (1994), Kennedy and Boguraev (1996), Denber (1998), and Vieira and Poesio (2000) are heuristic-based.
- *All of the features used (and hence the resulting classifier) are domain-independent.* Bean and Riloff (1999) present an unsupervised approach for constructing a list of non-anaphoric entities from an unannotated corpus. Although their approach is domain-independent, the strength of the approach lies in the use of lexical information for acquiring a list of non-anaphoric entities consisting mainly of domain-specific terms, which is useful for classifying documents from that domain.
- *The approach allows additional insight into the factors important to anaphoricity determination from an information-theoretic perspective.* In contrast, it is often not easy to evaluate the relative contribution of each knowledge source available to the problem with heuristic-based approaches (e.g. Paice and Husk (1987), Lappin and Leass (1994), Kennedy and Boguraev (1996), Denber (1998), Vieira and Poesio (2000)).
- *The approach performs anaphoricity determination on all types of NPs.* Some

[1] See van Deemter and Kibble (2000) for a discussion of the differences between anaphora and coreference.

anaphoricity determination algorithms address only pleonastic pronouns (e.g. Paice and Husk (1987), Lappin and Leass (1994), Kennedy and Boguraev (1996), Denber (1998)), while others focus on identifying anaphoric and non-anaphoric uses of *it* (e.g. Evans (2001)) and definite descriptions (e.g. Bean and Riloff (1999), Vieira and Poesio (2000)). In general, anaphoricity determination for non-definites is largely ignored by previous approaches.[2] We found, however, that 7% and 13% of the non-definite NPs in the MUC-6 and MUC-7 data sets (MUC-6, 1995; MUC-7, 1998), respectively, are anaphoric.

Second, we investigate empirically whether knowledge of anaphoricity can improve the performance of a learning-based coreference resolution system. Machine learning frameworks for coreference resolution typically recast the problem as a combination of classification and clustering. Specifically, pairs of NPs from a document are classified as co-referring or not based on constraints that are learned from an annotated corpus (e.g. Aone and Bennett (1995), Soon et al. (2001)); a separate clustering mechanism then coordinates the possibly contradictory pairwise classifications and constructs a partition on the set of NPs. Here we propose instead an architecture that employs an explicit anaphoricity determination component as a preprocessing filter for the coreference resolution component, both of which operate in conjunction with an on-line clustering algorithm for coordinating all anaphoricity and coreference decisions. Both the anaphoricity and coreference components are learning-based systems that can be trained on the same corpus.

Our results using anaphoricity determination for coreference resolution are mixed. In particular, we find that incorporating anaphoricity information causes performance to drop significantly when compared to the original coreference resolution system, as a result of large gains in precision but much larger drops in recall. We explain this phenomenon in terms of the (in)accuracy of the anaphoricity classifier and show that this drop is particularly dramatic for common noun phrases in contrast to pronouns and proper names. Augmenting the anaphoricity classifier with two heuristics that encode prior knowledge regarding common noun anaphoricity corrects this problem, and produces a coreference resolution system that outperforms the best MUC-6 and MUC-7 coreference systems.

The rest of the paper is organized as follows. In sections 2 and 3, we present the details of the anaphoricity determination and coreference resolution components, respectively. Section 4 describes and evaluates modifications to the coreference resolution component to incorporate (non-)anaphoricity information. Section 5 presents and evaluates two heuristics for improving performance of the anaphoricity classifier on common noun phrases. We conclude with future work in section 6.

2 The Anaphoricity Classifier

This section describes the anaphoricity determination component, which can be trained using standard supervised machine learning methods.

Building a classifier for anaphoricity determination. We use the C4.5 decision tree induction system (Quinlan, 1993) to train a classifier that, given a description of an NP in a document, decides whether or not the NP is anaphoric. Each training instance represents a single NP and consists of 37 features (see Table 1) that are potentially useful for distinguishing anaphoric and non-anaphoric NPs.[3]

Linguistically, the features can be divided into four groups: lexical, grammatical, semantic, and positional. The lexical features test whether some property holds for an NP based on its corresponding surface string. The grammatical features can be subcategorized into three groups. The first group simply determines the NP type, e.g. is the NP definite, or is it a pronoun? The second group tests some syntactic property of an NP (e.g. whether the NP is postmodified by a relative clause), largely aiming to detect non-anaphoric NPs. Similarly, the "syntactic pattern" features generally identify non-anaphoric definite NPs — definites without modifiers vs. definites modified by proper

[2]We consider an NP to be a non-definite if it is neither a pronoun nor a proper name and does not start with the article "the" or a demonstrative such as "this", "that", "these" or "those".

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

nouns. Instances also encode four semantic features, e.g. whether an NP is the TITLE of a person, or an ALIAS of a preceding NP. The final set of features makes anaphoricity decisions based on the position of the NP in the text.

The classification associated with a training instance — one of ANAPHORIC or NOT ANAPHORIC — is obtained from coreference chains in the MUC training texts. Specifically, a *positive instance* is created for each NP that is involved in a coreference chain but is not the head of the chain. A *negative instance* is created for each of the remaining NPs.

Applying the classifier. To determine the anaphoricity of an NP in a test document, we create an instance for it as during training and present it to the decision tree anaphoricity classifier, which returns a value of ANAPHORIC or NOT ANAPHORIC.

Evaluation. We evaluate the system using the MUC-6 (1995) and MUC-7 (1998) coreference data sets, training the anaphoricity classifier on the 30 "dry run" texts, and applying the classifier on the 20–30 "formal evaluation" texts. As a baseline measurement in which all test instances are classified as NOT ANAPHORIC (i.e. the majority class), we obtain an accuracy of 63.8% and 73.2% for the MUC-6 and MUC-7 data sets, respectively. The anaphoricity classifier, on the other hand, achieves accuracies of 86.1% and 84.0% for 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, in place of C4.5. Whether or not these performance levels are adequate for improving coreference resolution will be addressed in section 5.

Discussion. The resulting classifiers provide a convenient means for visualizing the features important for anaphoricity determination for the data sets investigated: the top portion of the pruned decision tree learned for the MUC-6 data set is shown in Figure 1. Of the 37 available features, the trees learned for the MUC-6 and MUC-7 data sets use only 14 and 15 features, respectively, having 9 features in common. In general, we see that HEAD_MATCH and ALIAS are important features for anaphoricity determination.

```
HEAD_MATCH = Y:
|  ALIAS = Y:
|  |  PROPER_NOUN = Y: + (408.0/16.2)
|  |  PROPER_NOUN = N:
|  |  |  EMBEDDED = Y: - (4.0/1.2)
|  |  |  EMBEDDED = N:
|  |  |  |  DEFINITE = Y: - (3.0/1.1)
|  |  |  |  DEFINITE = N: + (34.0/13.5)
|  ALIAS = N:
|  |  PRONOUN = Y: + (209.0/36.1)
|  |  PRONOUN = N:
|  |  |  THE_N = Y: + (186.0/35.2)
|  |  |  THE_N = N:
|  |  |  |  THE_PN_N = Y: + (8.0/2.4)
|  |  |  |  THE_PN_N = N:
|  |  |  |  |  POST = Y: + (11.0/3.6)
|  |  |  |  |  POST = N:
|  |  |  |  |  |
|  |  |  |  |  |
HEAD_MATCH = N:
|  PRONOUN = Y: + (105.0/27.7)
|  PRONOUN = N:
|  |  POST = Y:
|  |  |  PROPER_NOUN = Y: - (7.0/1.3)
|  |  |  PROPER_NOUN = N:
|  |  |  |  DEFINITE = Y: - (2.0/1.0)
|  |  |  |  DEFINITE = N: + (23.0/9.1)
|  |  POST = N:
|  |  |  THE_PN_N = Y: + (12.0/6.7)
|  |  |  THE_PN_N = N: - (2568.0/280.4)
```

Figure 1: Top portion of the decision tree for anaphoricity determination on the MUC-6 data set.

3 The Coreference Resolution System

This section gives an overview of the coreference resolution component, which combines classification and clustering to partition the NPs in a document into coreference equivalence classes.

Training an NP coreference classifier. We use C4.5 to train a classifier that, given a description of two NPs in a document, NP_i and NP_j, decides whether they are COREFERENT or NOT COREFERENT. Each training instance represents the two NPs under consideration and consists of 53 features. Like the anaphoricity classifier, the features for the coreference classifier can be divided into lexical, grammatical, semantic, and positional features. In contrast, however, features for the coreference classifier are tailored to the coreference resolution problem. We 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}, $\text{NP}_{i+2}, \ldots$, NP_{j-1}. Details of the coreference resolution component can be found in Ng and Cardie (2002); space limitations preclude their inclusion here.

Feature Type	Feature	Description
Lexical	STR_MATCH	Y if there exists a NP NP_i preceding NP_j such that, after discarding determiners, NP_i and NP_j are the same string; else N.
	HEAD_MATCH	Y if there exists an NP NP_i preceding NP_j such that NP_i and NP_j have the same head; else N.
	UPPERCASE	Y if NP_j is entirely in uppercase; else N.
	CONJ	Y if NP_j is a conjunction; else N.
Grammatical (NP type)	DEFINITE	Y if NP_j starts with "the"; else N.
	DEMONSTRATIVE	Y if NP_j starts with a demonstrative such as "this,", "that,", "these," or "those;" else N.
	INDEFINITE	Y if NP_j starts with "a" or "an"; else N.
	QUANTIFIED	Y if NP_j starts with quantifiers such as "every," "some," "all," "most," "many," "much," "few," or "none;" else N.
	ARTICLE	DEFINITE if NP_j is a definite NP; QUANTIFIED if NP_j is a quantified NP; else INDEFINITE.
	PRONOUN	Y if NP_j is a pronoun; else N.
	PROPER_NOUN	Y if NP_j is a proper noun; else N.
	POSSESSIVE	Y if NP_j starts with or is immediately preceded by a possessive pronoun or NP; else N.
	BARE_SINGULAR	Y if NP_j is singular and does not start with an article; else N.
	BARE_PLURAL	Y if NP_j is plural and does not start with an article; else N.
	EMBEDDED	Y if NP_j is a prenominal modifier; else N.
Grammatical (NP property/ relationship)	APPOSITIVE	Y if NP_j is the first of the two NPs in an appositive construction; else N.
	PREDNOM	Y if NP_j is the first of the two NPs in a predicate nominal construction; else N.
	NUMBER	SINGULAR if NP_j is singular in number; PLURAL if NP_j is plural in number; UNKNOWN if the number information cannot be determined.
	MODIFIER	Y if NP_j is premodified; else N.
	POSTMODIFIED	Y if NP_j is postmodified by a relative clause; else N.
	CONTAINS_PN	Y if NP_j is not a proper noun but contains a proper noun; else N.
	SPECIAL_NOUNS	Y if the head noun of NP_j is a comparative or NP_j is premodified by a superlative; else N.
Grammatical (Syntactic Pattern)	THE_N	Y if NP_j starts with "the" followed exactly by one common noun; else N.
	THE_2N	Y if NP_j starts with "the" followed exactly by two common nouns; else N.
	THE_PN	Y if NP_j starts with "the" followed exactly by a proper noun; else N.
	THE_PN_N	Y if NP_j starts with "the" followed exactly by a proper noun and a common noun; else N.
	THE_ADJ_N	Y if NP_j starts with "the" followed exactly by an adjective and a common noun; else N.
	THE_NUM_N	Y if NP_j starts with "the" followed exactly by a cardinal and a common noun; else N.
	THE_NE	Y if NP_j starts with "the" followed exactly by a named entity; else N.
	THE_SING_N	Y if NP_j starts with "the" followed a singular NP not containing any proper noun; else N.
Semantic	ALIAS	Y if there exists an NP NP_i preceding NP_j such that NP_i and NP_j are aliases; else N.
	POST	Y if NP_j is a post; else N.
	SUBCLASS	Y if there exists an NP NP_i preceding NP_jsuch that NP_i and NP_j have an ancestor-descendent relationship in WordNet; else N.
	TITLE	Y if NP_j is the title of a person; else N.
Positional	FIRST_SENT	Y if NP_j is in the first sentence of the body of the text; else N.
	FIRST_PARA	Y if NP_j is in the first paragraph of the body of the text; else N.
	HEADER	Y if NP_j is in the header of the text; else N.

Table 1: Feature Set for the Anaphoricity Determination System. Each instance represents a single NP, NP_j, characterized by 37 features.

Applying the classifier to create coreference chains. To apply the learned coreference classifier, texts are processed from left to right. Each NP encountered, NP_j, is compared in turn to each preceding NP, NP_i. 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.[4] The NP with the highest coreference likelihood value among the preceding NPs with coreference class values above 0.5 is selected as the antecedent of NP_j; otherwise, no antecedent is selected for NP_j.

[4]We convert the binary class value using a smoothed (Laplace) ratio of the number of positive instances to the total number of instances contained in leaf nodes.

Evaluation. We evaluate the baseline coreference system using the MUC-6 and MUC-7 coreference data sets, training the classifier on the dry run texts and applying it on the formal evaluation texts. Results are shown in the first row 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 system achieves an F-measure of 63.8 and 61.6 on the MUC-6 and MUC-7 data sets, respectively. With RIPPER, performance is better for the MUC-6 data set but worse for MUC-7.

The indented results show the performance of the system on pronouns, proper nouns and common nouns. One noticeable problem is the low precision on common noun resolution. We hypothesized that this is because the system mistakenly identifies an antecedent for many of non-anaphoric common nouns. In contrast to pronouns and proper nouns, 77.1% and 78.0% of the common nouns are non-anaphoric in the MUC-6 and MUC-7 data sets, respectively. With perfect anaphoricity information, for example, the coreference system achieves F-measures as high as 73.1 (for MUC-6) and 70.7 (for MUC-7) using C4.5 and an F-measure of 71.8 (for MUC-6) and 69.1 (for MUC-7) using RIPPER. As a result, we investigate the incorporation of anaphoricity information in the coreference resolution system in the next section.

4 Using Anaphoricity Information for Coreference Resolution

In this section, we augment the baseline coreference resolution system with an anaphoricity determination component.

System Architecture. The augmented system modifies only the clustering algorithm that imposes a partitioning on all NPs in the test texts to take into account information provided by the anaphoricity classifier: instead of comparing each NP encountered to each preceding NP in the text for potential merging, the clustering algorithm first uses the anaphoricity classifier to determine whether the NP under consideration is anaphoric. The NP is compared to preceding NPs only if it is considered anaphoric.

Results and Discussion. In general, we hypothesized that the precision on all types of NPs would increase if non-anaphoric NPs could be successfully identified by the anaphoricity classifier. In particular, the incorporation of anaphoricity information should be most beneficial to common nouns.

Results using the augmented system are shown in the second row of Table 2. The results largely support our hypothesis: we see significant improvement in precision for common nouns, which, however, comes at the expense of significant loss in recall. Similar trends are observed for proper nouns for the MUC-7 data set. Overall, we see statistically significant increases in precision, but much larger decreases in recall.[5] With one exception (MUC-6/C4.5), F-measure drops precipitously for both learning algorithms and both data sets.

A likely reason for the drop in recall especially for common nouns, and the decrease in overall performance with the augmented coreference system is the accuracy of the anaphoricity classifier on positive instances, i.e. NPs that are anaphoric. Specifically, if the classifier misclassifies an anaphoric entity as non-anaphoric, the entity will not be considered coreferent with any of its preceding NPs by the coreference system, potentially leading to a drop in recall. While Mitkov et al. (2002) also report that exploiting anaphoricity information precipitates a drop in the performance of their pronoun resolution system, they reiterate the usefulness of anaphoricity information in anaphora resolution and argue that the performance of the anaphoricity classifier is not directly captured by their scoring function for pronoun resolution. The MUC scoring program receives similar criticisms regarding its failure to directly reward successful identification of non-coreference relationships for which anaphoricity information can be useful (e.g. Bagga and Baldwin (1998)). Nevertheless, we believe that the drop in overall performance is ultimately caused by errors in anaphoricity determination and will attempt to remedy this problem in the next section.

[5] 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.

	C4.5						RIPPER					
	MUC-6			MUC-7			MUC-6			MUC-7		
Coref System Variation	R	P	F	R	P	F	R	P	F	R	P	F
Baseline (No Anaphor)	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	17.9	66.3	28.2	10.2	62.1	17.6	17.5	71.3	28.1	10.1	62.0	17.4
Proper nouns only	29.9	84.2	44.1	27.0	77.7	40.0	28.9	85.5	43.2	24.5	75.9	37.0
Common nouns only	25.2	40.1	31.0	26.6	45.2	33.5	23.3	43.7	30.4	25.7	48.0	33.5
Anaphor (No Constraints)	57.4	71.6	63.7	47.0	77.1	58.4	56.7	70.5	62.8	45.9	75.3	57.0
Pronouns only	17.9	67.0	28.2	10.2	62.1	17.6	17.5	71.3	28.1	10.1	62.0	17.4
Proper Nouns only	26.6	89.2	41.0	21.5	84.8	34.3	26.2	89.0	40.6	21.6	84.0	34.2
Common Nouns only	15.4	56.2	24.2	13.8	77.5	23.4	15.5	52.2	23.9	13.4	73.7	22.7
Anaphor (Constraints)	63.4	68.3	**65.8**	59.7	69.3	**64.2**	61.2	69.6	**65.1**	56.9	70.4	**62.9**
Pronouns only	17.9	67.0	28.2	10.2	62.1	17.6	17.5	71.3	28.1	10.1	62.0	17.4
Proper Nouns only	27.4	88.5	41.9	26.1	84.7	40.0	27.0	88.9	41.4	25.0	85.3	38.7
Common Nouns only	20.5	53.1	29.6	21.7	59.0	31.7	18.7	52.9	27.6	20.8	60.9	31.0

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.

5 Classification with Constraints

Although overall accuracy of the anaphoricity classifier is 86.1% and 84.0% for the MUC-6 and MUC-7 data sets, respectively, accuracy on just the **positive** instances is substantially lower — 73.1% and 66.2%. In particular, the classifier misclassifies 414 and 322 anaphoric entities as non-anaphoric for the MUC-6 and MUC-7 data sets, respectively. Evidently, there is room for improvement. In particular, a closer examination of the baseline coreference classifiers in section 3 (not shown) reveals that string matching and aliasing are strong indicators of coreference.

It is possible then that the drop in recall for coreference resolution of common nouns and proper nouns is attributable to the anaphoricity classifier's misclassifications of anaphoric NPs involved in these two types of relations. Consequently, we next augment the coreference resolution component to first apply the STR_MATCH and ALIAS constraints (both of which are identical to the features with the same names in the anaphoricity classifier). If anaphoricity is indicated by either constraint, the NP is assumed to be anaphoric and the anaphoricity classifier is bypassed. The two constraints essentially increase the number of entities classified as anaphoric, thus potentially enhancing the accuracy of the classifier on the positive instances.

Results and Discussion. The addition of constraints provides statistically significant increases in accuracy on truly anaphoric instances, from 73.1% to 82.0% for MUC-6 and from 66.2% to 80.8% for MUC-7 using C4.5. Similar results are obtained using RIPPER. The performance of the coreference system using the anaphoricity information generated with constraints is shown in the third row of Table 2. As expected, in comparison to the baseline coreference system, we see statistically significant gains in precision, and smaller drops in recall than were observed with the original anaphoricity component; recall levels for common nouns also increase. Furthermore, the resulting F-measure scores are higher than those produced by the best-performing MUC systems on the corresponding coreference data sets: F-measure increases w.r.t. the baseline coreference system from 63.8 to 65.8 for MUC-6/C4.5, and from 61.6 to 64.2 for MUC-7/C4.5. Overall, our results support the hypothesis that automatically acquired knowledge of anaphoricity can be used to improve coreference resolution, provided that such information can be learned accurately. In particular, improvements stem from increases in accuracy on negative instances, and are most pronounced for common nouns. Nevertheless, our coreference resolution results using perfect anaphoricity information (end of section 3) posted F-measures that are 6-7% better still, indicating that there remains substantial room for improvement in anaphoricity determination.

6 Conclusions and Future Work

We have presented a supervised learning approach for anaphoricity determination that can handle all types of NPs. In addition, we have investigated whether knowledge of anaphoricity can improve the performance of a learning-based coreference resolution system by proposing an architecture that employs an explicit anaphoricity determination component as a preprocessing filter for the coreference resolution component. We have shown that coreference resolution systems can improve by making use of anaphoricity information, outperforming the best-performing MUC-6 and MUC-7 coreference resolution systems on the corresponding coreference data sets — obtaining F-measures of 65.8 and 64.2, respectively.

The approach, however, can be improved in a number of ways. In particular, we will investigate whether the accuracy of classifying non-anaphoric entities can be improved without resorting to the use of ad-hoc constraints for post-processing the output of the anaphoricity classifier. To this end, we plan to experiment with cost-sensitive learning algorithms (e.g. Turney (1995)) that provide the flexibility of adjusting misclassification costs on examples of different classes. Nevertheless, the use of anaphoricity information for coreference resolution constitutes a promising computational approach that can potentially reduce a coreference system's reliance on sophisticated world knowledge.

Acknowledgments

Thanks to three anonymous reviewers for their comments. 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. Bagga and B. Baldwin. 1998. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and COLING-98*, pages 79–85.

D. Bean and E. Riloff. 1999. Corpus-Based Identification of Non-Anaphoric Noun Phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 373–380.

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

M. Denber. 1998. Automatic Resolution of Anaphora in English. Technical report, Eastman Kodak Co.

R. Evans. 2001. Applying Machine Learning toward an Automatic Classification of it. *Literary and Linguistic Computing*, 16(1):45–57.

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*, pages 55–62.

C. Kennedy and B. Boguraev. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *Proceedings of the 16th International Conference on Computational Linguistics.*

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

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

R. Mitkov, R. Evans, and C. Orasan. 2002. A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method. In Al. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 169–187. Springer.

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.*

C. Paice and G. Husk. 1987. Towards the Automatic Recognition of Anaphoric Features in English Text: the Impersonal Pronoun 'it'. *Computer Speech and Language*, 2.

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

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.

P. Turney. 1995. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *JAIR*, 2:369–409.

K. van Deemter and R. Kibble. 2000. On Coreferring: Coreference in MUC and related annotation schemes. *Computational Linguistics*, 26(4).

R. Vieira and M. Poesio. 2000. Processing definite descriptions in corpora. In S. Botley and A. McEnery, editors, *Corpus-based and Computational Approaches to Discourse Anaphora.*

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.