Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 82–89, Association for Computational Linguistics, 1999.

Noun Phrase Coreference as Clustering

Claire Cardie and Kiri Wagstaff
Department of Computer Science
Cornell University
Ithaca, NY 14853
E-mail: cardie,wkiri@cs.cornell.edu

Abstract

This paper introduces a new, unsupervised algorithm for noun phrase coreference resolution. It differs from existing methods in that it views coreference resolution as a clustering task. In an evaluation on the MUC-6 coreference resolution corpus, the algorithm achieves an F-measure of 53.6%, placing it firmly between the worst (40%) and best (65%) systems in the MUC-6 evaluation. More importantly, the clustering approach outperforms the only MUC-6 system to treat coreference resolution as a learning problem. The clustering algorithm appears to provide a flexible mechanism for coordinating the application of context-independent and context-dependent constraints and preferences for accurate partitioning of noun phrases into coreference equivalence classes.

1 Introduction

Many natural language processing (NLP) applications require accurate noun phrase coreference resolution: They require a means for determining which noun phrases in a text or dialogue refer to the same real-world entity. The vast majority of algorithms for noun phrase coreference combine syntactic and, less often, semantic cues via a set of hand-crafted heuristics and filters. All but one system in the MUC-6 coreference performance evaluation (MUC-6, 1995), for example, handled coreference resolution in this manner. This same reliance on complicated hand-crafted algorithms is true even for the narrower task of pronoun resolution. Some exceptions exist, however. Ge et al. (1998) present a probabilistic model for pronoun resolution trained on a small subset of the Penn Treebank Wall Street Journal corpus (Marcus et al., 1993). Dagan and Itai (1991) develop a statistical filter for resolution of the pronoun "it" that selects among syntactically viable antecedents based on relevant subject-verb-object cooccurrences. Aone and Bennett (1995) and McCarthy and Lehnert (1995) employ decision tree algorithms to handle a broader subset of general noun phrase coreference problems.

This paper presents a new corpus-based approach to noun phrase coreference. We believe that it is the first such unsupervised technique developed for the general noun phrase coreference task. In short, we view the task of noun phrase coreference resolution as a clustering task. First, each noun phrase in a document is represented as a vector of attribute-value pairs. Given the feature vector for each noun phrase, the clustering algorithm coordinates the application of context-independent and context-dependent coreference constraints and preferences to partition the noun phrases into equivalence classes, one class for each real-world entity mentioned in the text. Context-independent coreference constraints and preferences are those that apply to two noun phrases in isolation. Context-dependent coreference decisions, on the other hand, consider the relationship of each noun phrase to surrounding noun phrases.

In an evaluation on the MUC-6 coreference resolution corpus, our clustering approach achieves an F-measure of 53.6%, placing it firmly between the worst (40%) and best (65%) systems in the MUC-6 evaluation. More importantly, the clustering approach outperforms the only MUC-6 system to view coreference resolution as a learning problem: The RESOLVE system (McCarthy and Lehnert, 1995) employs decision tree induction and achieves an F-measure of 47% on the MUC-6 data set. Furthermore, our approach has a number of important advantages over existing learning and non-learning methods for coreference resolution:

- The approach is largely unsupervised, so no annotated training corpus is required.
- Although evaluated in an information ex-

traction context, the approach is domain-independent.

- As noted above, the clustering approach provides a flexible mechanism for coordinating context-independent and context-dependent coreference constraints and preferences for partitioning noun phrases into coreference equivalence classes.

As a result, we believe that viewing noun phrase coreference as clustering provides a promising framework for corpus-based coreference resolution.

The remainder of the paper describes the details of our approach. The next section provides a concrete specification of the noun phrase coreference resolution task. Section 3 presents the clustering algorithm. Evaluation of the approach appears in Section 4. Qualitative and quantitative comparisons to related work are included in Section 5.

2 Noun Phrase Coreference

It is commonly observed that a human speaker or author avoids repetition by using a variety of noun phrases to refer to the same entity. While human audiences have little trouble mapping a collection of noun phrases onto the same entity, this task of *noun phrase (NP) coreference resolution* can present a formidable challenge to an NLP system. Figure 1 depicts a typical coreference resolution system, which takes as input an arbitrary document and produces as output the appropriate coreference equivalence classes. The subscripted noun phrases in the sample output constitute two noun phrase coreference equivalence classes: Class JS contains the five noun phrases that refer to John Simon, and class PC contains the two noun phrases that represent Prime Corp. The figure also visually links neighboring coreferent noun phrases. The remaining (unbracketed) noun phrases have no coreferent NPs and are considered singleton equivalence classes. Handling the JS class alone requires recognizing coreferent NPs in appositive and genitive constructions as well as those that occur as proper names, possessive pronouns, and definite NPs.

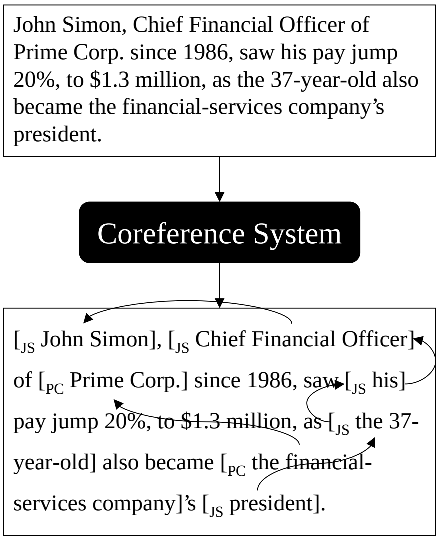

Figure 1: Coreference System

3 Coreference as Clustering

Our approach to the coreference task stems from the observation that each group of coreferent noun phrases defines an equivalence class[1]. Therefore, it is natural to view the problem as one of partitioning, or clustering, the noun phrases. Intuitively, all of the noun phrases used to describe a specific concept will be "near" or related in some way, i.e. their conceptual "distance" will be small. Given a description of each noun phrase and a method for measuring the distance between two noun phrases, a clustering algorithm can then group noun phrases together: Noun phrases with distance greater than a clustering radius r are not placed into the same partition and so are not considered coreferent.

The subsections below describe the noun phrase representation, the distance metric, and the clustering algorithm in turn.

3.1 Instance Representation

Given an input text, we first use the Empire noun phrase finder (Cardie and Pierce, 1998) to locate all noun phrases in the text. Note that Empire identifies only *base noun phrases*, i.e. simple noun phrases that contain no other smaller noun phrases within them. For example, *Chief Financial Officer of Prime Corp.* is too complex to be a base noun phrase. It contains two base noun phrases: *Chief Financial Officer* and *Prime Corp.*

Each noun phrase in the input text is then represented as a set of 11 features as shown in Table 1. This noun phrase representation is a first approximation to the feature vector that would be required for accurate coreference resolution. All feature values are automatically generated and, therefore, are not always perfect. In particular, we use very simple heuristics to approximate the behavior of more complex feature value computations:

Individual Words. The words contained in the

[1]The coreference relation is symmetric, transitive, and reflexive.

Words, Head Noun (in **bold**)	Position	Pronoun Type	Article	Appositive	Number	Proper Name	Semantic Class	Gender	Animacy
John **Simon**	1	NONE	NONE	NO	SING	YES	HUMAN	MASC	ANIM
Chief Financial **Officer**	2	NONE	NONE	NO	SING	NO	HUMAN	EITHER	ANIM
Prime **Corp.**	3	NONE	NONE	NO	SING	NO	COMPANY	NEUTER	INANIM
1986	4	NONE	NONE	NO	PLURAL	NO	NUMBER	NEUTER	INANIM
his	5	POSS	NONE	NO	SING	NO	HUMAN	MASC	ANIM
pay	6	NONE	NONE	NO	SING	NO	PAYMENT	NEUTER	INANIM
20%	7	NONE	NONE	NO	PLURAL	NO	PERCENT	NEUTER	INANIM
$1.3 **million**	8	NONE	NONE	NO	PLURAL	NO	MONEY	NEUTER	INANIM
the **37-year-old**	9	NONE	DEF	NO	SING	NO	HUMAN	EITHER	ANIM
the financial-services **company**	10	NONE	DEF	NO	SING	NO	COMPANY	NEUTER	INANIM
president	11	NONE	NONE	NO	SING	NO	HUMAN	EITHER	ANIM

Table 1: Noun Phrase Instance Representation For All Base NPs in the Sample Text

noun phrase are stored as a feature.

Head noun. The last word in the noun phrase is considered the head noun.

Position. Noun phrases are numbered sequentially, starting at the beginning of the document.

Pronoun Type. Pronouns are marked as one of NOMinative, ACCusative, POSSessive, or AMBiguous (*you* and *it*). All other noun phrases obtain the value NONE for this feature.

Article. Each noun phrase is marked INDEFinite (contains *a* or *an*), DEFinite (contains *the*), or NONE.

Appositive. Here we use a simple, overly restrictive heuristic to determine whether or not the noun phrase is in a (post-posed) appositive construction: If the noun phrase is surrounded by commas, contains an article, and is immediately preceded by another noun phrase, then it is marked as an appositive.

Number. If the head noun ends in an 's', then the noun phrase is marked PLURAL; otherwise, it is considered SINGular. Expressions denoting money, numbers, or percentages are also marked as PLURAL.

Proper Name. Proper names are identified by looking for two adjacent capitalized words, optionally containing a middle initial.

Semantic Class. Here we use WordNet (Fellbaum, 1998) to obtain coarse semantic information for the head noun. The head noun is characterized as one of TIME, CITY, ANIMAL, HUMAN, or OBJECT. If none of these classes pertains to the head noun, its immediate parent in the class hierarchy is returned as the semantic class, e.g. PAYMENT for the head noun *pay* in NP_6 of Table 1. A separate algorithm identifies NUMBERS, MONEY, and COMPANYs.

Gender. Gender (MASCuline, FEMinine, EITHER, or NEUTER) is determined using WordNet and (for proper names) a list of common first names.

Animacy. Noun phrases classified as HUMAN or ANIMAL are marked ANIM; all other NPs are considered INANIM.

3.2 Distance Metric

Next, we define the following distance metric between two noun phrases:

$$dist(NP_i, NP_j) = \sum_{f \in F} w_f * incompatibility_f(NP_i, NP_j)$$

where F corresponds to the NP feature set described above; $incompatibility_f$ is a function that returns a value between 0 and 1 inclusive and indicates the degree of incompatibility of f for NP_i and NP_j; and w_f denotes the relative importance of compatibility w.r.t. feature f. The incompatibility functions and corresponding weights are listed in Table 2.[2] In general, weights are chosen to represent linguistic knowledge about coreference. Terms with a weight of ∞ represent filters that rule out impossible antecedents: Two noun phrases can *never* corefer when they have incompatible values for that term's feature. In the current version of our system, the NUMBER, PROPER NAME, SEMANTIC CLASS, GENDER, and ANIMACY features operate as coreference filters. Conversely, terms with a weight of $-\infty$ force coreference between two noun phrases with compatible values for that term's feature. The APPOSITIVE and WORDS-SUBSTRING terms operate in this fashion in the current distance metric.

Terms with a weight of r — the clustering radius threshold — implement a preference that two NPs not be coreferent if they are incompatible w.r.t. that term's feature. As will be explained below,

[2] Note that there is not currently a one-to-one correspondence between NP features and distance metric terms: The distance metric contains two terms that make use of the WORDS feature of the noun phrase representation.

Feature f	Weight	Incompatibility function
Words	10.0	(# of mismatching words[a]) / (# of words in the longer NP)
Head Noun	1.0	1 if the head nouns differ; else 0
Position	5.0	(difference in position) / (maximum difference in document)
Pronoun	r	1 if NP_i is a pronoun and NP_j is not; else 0
Article	r	1 if NP_j is indefinite and not appositive; else 0
Words–Substring	$-\infty$	1 if NP_i subsumes (entirely includes as a substring) NP_j;
Appositive	$-\infty$	1 if NP_j is appositive and NP_i is its immediate predecessor; else 0
Number	∞	1 if they do not match in number; else 0
Proper Name	∞	1 if both are proper names, but mismatch on every word; else 0
Semantic Class	∞	1 if they do not match in class; else 0
Gender	∞	1 if they do not match in gender (allows EITHER to match MASC or FEM); else 0
Animacy	∞	1 if they do not match in animacy; else 0

[a]Pronouns are handled as gender-specific "wild cards".

Table 2: Incompatibility Functions and Weights for Each Term in the Distance Metric

Words, **Head Noun**	Position	Pronoun	Article	Appositive	Number	Proper Name	Class	Gender	Animacy
The **chairman**	1	NONE	DEF	NO	SING	NO	HUMAN	EITHER	ANIM
Ms. **White**	2	NONE	NONE	NO	SING	YES	HUMAN	FEM	ANIM
He	3	NOM	NONE	NO	SING	NO	HUMAN	MASC	ANIM

Table 3: Instance Representation for Noun Phrases in *The chairman spoke with Ms. White yesterday. He...*

however, two such NPs can be merged into the same equivalence class by the clustering algorithm if there is enough other evidence that they are similar (i.e. there are other, coreferent noun phrase(s) sufficiently close to both).

All other terms obtain weights selected using the development corpus. Although additional testing is required, our current results indicate that these weights are sensitive to the distance metric, but probably not to the corpus.

When computing a sum that involves both ∞ and $-\infty$, we choose the more conservative route, and the ∞ distance takes precedence (i.e. the two noun phrases are not considered coreferent). An example of where this might occur is in the following sentence:

[$_1$ *Reardon Steel Co.*] *manufactures several thousand tons of* [$_2$ *steel*] *each week.*

Here, NP_1 subsumes NP_2, giving them a distance of $-\infty$ via the word substring term; however, NP_1's semantic class is COMPANY, and NP_2's class is OBJECT, generating a distance of ∞ via the semantic class feature. Therefore, $dist(NP_1, NP_2) = \infty$ and the two noun phrases are not considered coreferent.

The coreference distance metric is largely context-independent in that it determines the distance between two noun phrases using very little, if any, of their intervening or surrounding context. The clustering algorithm described below is responsible for coordinating these local coreference decisions across arbitrarily long contexts and, thus, implements a series of context-dependent coreference decisions.

3.3 Clustering Algorithm

The clustering algorithm is given in Figure 2. Because noun phrases generally refer to noun phrases that precede them, we start at the end of the document and work backwards. Each noun phrase is compared to all preceding noun phrases. If the distance between two noun phrases is less than the clustering radius r, then their classes are considered for possible merging. Two coreference equivalence classes can be merged unless there exist any incompatible NPs in the classes to be merged.

It is useful to consider the application of our algorithm to an excerpt from a document:

[$_1$ *The chairman*] *spoke with* [$_2$ *Ms. White*] *yesterday.* [$_3$ *He*] *...*

The noun phrase instances for this fragment are shown in Table 3. Initially, NP_1, NP_2, and NP_3 are all singletons and belong to coreference classes c_1, c_2, and c_3, respectively. We begin by considering NP_3. $Dist(NP_2, NP_3) = \infty$ due to a mismatch on gender, so they are not considered for possible merging. Next, we calculate the distance from NP_1 to NP_3. Pronouns are not expected to match when the words of two noun phrases are compared, so there is no penalty here for word (or head noun) mismatches. The penalty for their difference

---

Coreference_Clustering $(NP_n, NP_{n-1}, \ldots, NP_1)$

1. Let r be the clustering radius.
2. Mark each noun phrase NP_i as belonging to its own class, c_i: $c_i = \{NP_i\}$.
3. Proceed through the noun phrases from the document *in reverse order*, $NP_n, NP_{n-1}, \ldots, NP_1$. For each noun phrase NP_j encountered, consider each preceding noun phrase NP_i.
   (a) Let $d = dist(NP_i, NP_j)$.
   (b) Let $c_i = class\_ofNP_i$ and $c_j = class\_ofNP_j$.
   (c) If $d < r$ and All_NPs_Compatible (c_i, c_j) then $c_j = c_i \cup c_j$.

All_NPs_Compatible (c_i, c_j)

1. For all $NP_a \in c_j$
   (a) For all $NP_b \in c_i$
      i. If $dist(NP_a, NP_b) = \infty$ then Return false.
2. Return true.

Figure 2: Clustering Algorithm

in position is dependent on the length of the document. For illustration, assume that this is less than r. Thus, $dist(NP_1, NP_3) < r$. Their coreference classes, c_1 and c_3, are then considered for merging. Because they are singleton classes, there is no additional possibility for conflict, and both noun phrases are merged into c_1.

The algorithm then considers NP_2. $Dist(NP_1, NP_2) = 11.0$ plus a small penalty for their difference in position. If this distance is $\geq r$, they will not be considered coreferent, and the resulting equivalence classes will be: {{The chairman, he}, {Ms. White}}. Otherwise, the distance is $< r$, and the algorithm considers c_1 and c_2 for merging. However, c_1 contains NP_3, and, as calculated above, the distance from NP_2 to NP_3 is ∞. This incompatibility prevents the merging of c_1 and c_2, so the resulting equivalence classes would still be {{The chairman, he}, {Ms. White}}.

In this way, the equivalence classes grow in a flexible manner. In particular, the clustering algorithm automatically computes the transitive closure of the coreference relation. For instance, if $dist(NP_i, NP_j) < r$ and $dist(NP_j, NP_k) < r$ then (assuming no incompatible NPs), NP_i, NP_j, and NP_k will be in the same class and considered mutually coreferent. In fact, it is possible that $dist(NP_i, NP_k) \geq r$, according to the distance measure; but as long as that distance is not ∞, NP_i can be in the same class as NP_k. The distance measure operates on two noun phrases in isolation, but the clustering algorithm can and does make use of intervening NP information: intervening noun phrases can form a chain that links otherwise distant NPs. By separating context-independent and context-dependent computations, the noun phrase representation and distance metric can remain fairly simple and easily extensible as additional knowledge sources are made available to the NLP system for coreference resolution.

4 Evaluation

We developed and evaluated the clustering approach to coreference resolution using the "dry run" and "formal evaluation" MUC-6 coreference corpora. Each corpus contains 30 documents that have been annotated with NP coreference links. We used the dryrun data for development of the distance measure and selection of the clustering radius r and reserved the formal evaluation materials for testing. All results are reported using the standard measures of recall and precision or F-measure (which combines recall and precision equally). They were calculated automatically using the MUC-6 scoring program (Vilain et al., 1995).

Table 4 summarizes our results and compares them to three baselines. For each algorithm, we show the F-measure for the dryrun evaluation (column 2) and the formal evaluation (column 4). (The "adjusted" results are described below.) For the dryrun data set, the clustering algorithm obtains 48.8% recall and 57.4% precision. The formal evaluation produces similar scores: 52.7% recall and 54.6% precision. Both runs use $r = 4$, which was obtained by testing different values on the dryrun corpus. Table 5 summarizes the results on the dryrun data set for r values from 1.0 to 10.0.[3] As expected, increasing r also increases recall, but decreases precision. Subsequent tests with different values for r on the formal evaluation data set also obtained optimal performance with $r = 4$. This provides partial support for our hypothesis that r need not be recalculated for new corpora.

The remaining rows in Table 4 show the performance of the three baseline algorithms. The first baseline marks every pair of noun phrases as coreferent, i.e. all noun phrases in the document form one class. This baseline is useful because it establishes an upper bound for recall on our clustering algorithm (67% for the dryrun and 69% for the formal evaluation). The second baseline marks as corefer-

[3]Note that r need not be an integer, especially when the distance metric is returning non-integral values.

Algorithm	Dryrun Data Set		Formal Run Data Set	
	Official	Adjusted	Official	Adjusted
Clustering	**52.8**	**64.9**	**53.6**	**63.5**
All One Class	44.8	50.2	41.5	45.7
Match Any Word	44.1	52.8	41.3	48.8
Match Head Noun	46.5	56.9	45.7	54.9

Table 4: F-measure Results for the Clustering Algorithm and Baseline Systems on the MUC-6 Data Sets

ent any two noun phrases that have a word in common. The third baseline marks as coreferent any two noun phrases whose head nouns match. Although the baselines perform better one might expect (they outperform one MUC-6 system), the clustering algorithm performs significantly better.

In part because we rely on base noun phrases, our recall levels are fairly low. The "adjusted" figures of Table 4 reflect this upper bound on recall. Considering only coreference links between base noun phrases, the clustering algorithm obtains a recall of 72.4% on the dryrun, and 75.9% on the formal evaluation. Another source of error is inaccurate and inadequate NP feature vectors. Our procedure for computing semantic class values, for example, is responsible for many errors — it sometimes returns incorrect values and the coarse semantic class distinctions are often inadequate. Without a better named entity finder, computing feature vectors for proper nouns is difficult. Other errors result from a lack of thematic and grammatical role information. The lack of discourse-related topic and focus information also limits system performance. In addition, we currently make no special attempt to handle reflexive pronouns and pleonastic "it".

Lastly, errors arise from the greedy nature of the clustering algorithm. Noun phrase NP_j is linked to every preceding noun phrase NP_i that is compatible and within the radius r, and that link can never be undone. We are considering three possible ways to make the algorithm less aggressively greedy. First, for each NP_j, instead of considering every previous noun phrase, the algorithm could stop on finding the first compatible antecedent. Second, for each NP_j, the algorithm could rank all possible antecedents and then choose the best one and link only to that one. Lastly, the algorithm could rank all possible coreference links (all pairs of noun phrases in the document) and then proceed through them in ranked order, thus progressing from the links it is most confident about to those it is less certain of. Future work will include a more detailed error analysis.

r	Recall	Precision	F-measure
1	34.6	69.3	46.1
2	44.7	61.4	51.7
3	47.3	58.5	52.3
4	**48.8**	**57.4**	**52.8**
5	49.1	56.8	52.7
6	49.8	55.0	52.3
7	50.3	53.8	52.0
8	50.7	53.0	51.8
9	50.9	52.5	51.7
10	50.9	52.1	51.5

Table 5: Performance on the Dryrun Data Set for Different r

5 Related Work

Existing systems for noun phrase coreference resolution can be broadly characterized as learning and non-learning approaches. All previous attempts to view coreference as a learning problem treat coreference resolution as a classification task: the algorithms classify a pair of noun phrases as coreferent or not. Both MLR (Aone and Bennett, 1995) and RESOLVE (McCarthy and Lehnert, 1995), for example, apply the C4.5 decision tree induction algorithm (Quinlan, 1993) to the task. As supervised learning algorithms, both systems require a fairly large amount of training data that has been annotated with coreference resolution information. Our approach, on the other hand, uses unsupervised learning[4] and requires no training data.[5] In addition, both MLR and RESOLVE require an additional mechanism to coordinate the collection of pairwise coreference decisions. Without this mechanism, it is possible that the decision tree classifies NP_i and NP_j as coreferent, and NP_j and NP_k as coreferent, but NP_i and NP_k as *not* coreferent. In an evaluation on the MUC-6 data set (see Table 6), RESOLVE achieves an F-measure of 47%.

The MUC-6 evaluation also provided results for a

[4] Whether or not clustering can be considered a "learning" approach is unclear. The algorithm uses the existing partitions to process each successive NP, but the partitions generated for a document are not useful for processing subsequent documents.

[5] We do use training data to tune r, but as noted above, it is likely that r need not be recalculated for new corpora.

Algorithm	Recall	Precision	F-measure
Clustering	**53**	**55**	**54**
RESOLVE	44	51	47
Best MUC-6	59	72	65
Worst MUC-6	36	44	40

Table 6: Results on the MUC-6 Formal Evaluation

large number of non-learning approaches to coreference resolution. Table 6 provides a comparison of our results to the best and worst of these systems. Most implemented a series of linguistic constraints similar in spirit to those employed in our system. The main advantage of our approach is that all constraints and preferences are represented neatly in the distance metric (and radius r), allowing for simple modification of this measure to incorporate new knowledge sources. In addition, we anticipate being able to automatically learn the weights used in the distance metric.

There is also a growing body of work on the narrower task of pronoun resolution. Azzam et al. (1998), for example, describe a focus-based approach that incorporates discourse information when resolving pronouns. Lappin and Leass (1994) make use of a series of filters to rule out impossible antecedents, many of which are similar to our ∞-incompatibilities. They also make use of more extensive syntactic information (such as the thematic role each noun phrase plays), and thus require a fuller parse of the input text. Ge et al. (1998) present a supervised probabilistic algorithm that assumes a full parse of the input text. Dagan and Itai (1991) present a hybrid full-parse/unsupervised learning approach that focuses on resolving "it". Despite a large corpus (150 million words), their approach suffers from sparse data problems, but works well when enough relevant data is available. Lastly, Cardie (1992a; 1992b) presents a case-based learning approach for relative pronoun disambiguation.

Our clustering approach differs from this previous work in several ways. First, because we only require the noun phrases in any input text, we do not require a full syntactic parse. Although we would expect increases in performance if complex noun phrases were used, our restriction to base NPs does not reflect a limitation of the clustering algorithm (or the distance metric), but rather a self-imposed limitation on the preprocessing requirements of the approach. Second, our approach is unsupervised and requires no annotation of training data, nor a large corpus for computing statistical occurrences. Finally, we handle a wide array of noun phrase coreference, beyond just pronoun resolution.

6 Conclusions and Future Work

We have presented a new approach to noun phrase coreference resolution that treats the problem as a clustering task. In an evaluation on the MUC-6 coreference resolution data set, the approach achieves very promising results, outperforming the only other corpus-based learning approach and producing recall and precision scores that place it firmly between the best and worst coreference systems in the evaluation. In contrast to other approaches to coreference resolution, ours is unsupervised and offers several potential advantages over existing methods: no annotated training data is required, the distance metric can be easily extended to account for additional linguistic information as it becomes available to the NLP system, and the clustering approach provides a flexible mechanism for combining a variety of constraints and preferences to impose a partitioning on the noun phrases in a text into coreference equivalence classes.

Nevertheless, the approach can be improved in a number of ways. Additional analysis and evaluation on new corpora are required to determine the generality of the approach. Our current distance metric and noun phrase instance representation are only first, and admittedly very coarse, approximations to those ultimately required for handling the wide variety of anaphoric expressions that comprise noun phrase coreference. We would also like to make use of cues from centering theory and plan to explore the possibility of learning the weights associated with each term in the distance metric. Our methods for producing the noun phrase feature vector are also overly simplistic. Nevertheless, the relatively strong performance of the technique indicates that clustering constitutes a powerful and natural approach to noun phrase coreference resolution.

7 Acknowledgments

This work was supported in part by NSF Grants IRI–9624639 and EIA 97–03470, and a National Science Foundation Graduate fellowship. We would like to thank David Pierce for his formatting and technical advice.

References

Chinatsu Aone and William 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. Association for Computational Linguistics.

S. Azzam, K. Humphreys, and R. Gaizauskas. 1998. Evaluating a Focus-Based Approach to Anaphora Resolution. In *Proceedings of the 36th Annual*

Meeting of the Association for Computational Linguistics and COLING-98, pages 74–78. Association for Computational Linguistics.

C. Cardie and D. Pierce. 1998. Error-Driven Pruning of Treebank Grammars for Base Noun Phrase Identification. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and COLING-98*, pages 218–224, University of Montreal, Montreal, Canada. Association for Computational Linguistics.

C. Cardie. 1992a. Corpus-Based Acquisition of Relative Pronoun Disambiguation Heuristics. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 216–223, University of Delaware, Newark, DE. Association for Computational Linguistics.

C. Cardie. 1992b. Learning to Disambiguate Relative Pronouns. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 38–43, San Jose, CA. AAAI Press / MIT Press.

I. Dagan and A. Itai. 1991. A Statistical Filter for Resolving Pronoun References. In Y. A. Feldman and A. Bruckstein, editors, *Artificial Intelligence and Computer Vision*, pages 125–135. Elsevier Science Publishers, North Holland.

C. Fellbaum. 1998. *WordNet: An Electronical Lexical Database*. MIT Press, Cambridge, MA.

N. Ge, J. Hale, and E. Charniak. 1998. A Statistical Approach to Anaphora Resolution. In Charniak, Eugene, editor, *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–170, Montreal, Canada. ACL SIGDAT.

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

M. Marcus, M. Marcinkiewicz, and B. Santorini. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

J. McCarthy and W. Lehnert. 1995. Using Decision Trees for Coreference Resolution. In C. Mellish, editor, *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.

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

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.