A Machine Learning Approach to Coreference Resolution of Noun Phrases

Wee Meng Soon*
DSO National Laboratories

Hwee Tou Ng†
DSO National Laboratories

Daniel Chung Yong Lim‡
DSO National Laboratories

In this paper, we present a learning approach to coreference resolution of noun phrases in unrestricted text. The approach learns from a small, annotated corpus and the task includes resolving not just a certain type of noun phrase (e.g., pronouns) but rather general noun phrases. It also does not restrict the entity types of the noun phrases; that is, coreference is assigned whether they are of "organization," "person," or other types. We evaluate our approach on common data sets (namely, the MUC-6 and MUC-7 coreference corpora) and obtain encouraging results, indicating that on the general noun phrase coreference task, the learning approach holds promise and achieves accuracy comparable to that of nonlearning approaches. Our system is the first learning-based system that offers performance comparable to that of state-of-the-art nonlearning systems on these data sets.

1. Introduction

Coreference resolution is the process of determining whether two expressions in natural language refer to the same entity in the world. It is an important subtask in natural language processing systems. In particular, information extraction (IE) systems like those built in the DARPA Message Understanding Conferences (Chinchor 1998; Sundheim 1995) have revealed that coreference resolution is such a critical component of IE systems that a separate coreference subtask has been defined and evaluated since MUC-6 (MUC-6 1995).

In this paper, we focus on the task of determining coreference relations as defined in MUC-6 (MUC-6 1995) and MUC-7 (MUC-7 1997). Specifically, a coreference relation denotes an identity of reference and holds between two textual elements known as markables, which can be definite noun phrases, demonstrative noun phrases, proper names, appositives, sub–noun phrases that act as modifiers, pronouns, and so on. Thus, our coreference task resolves general noun phrases and is not restricted to a certain type of noun phrase such as pronouns. Also, we do not place any restriction on the possible candidate markables; that is, all markables, whether they are "organization," "person," or other entity types, are considered. The ability to link coreferring noun phrases both within and across sentences is critical to discourse analysis and language understanding in general.

* 20 Science Park Drive, Singapore 118230. E-mail: sweemeng@dso.org.sg
† 20 Science Park Drive, Singapore 118230. E-mail: nhweetou@dso.org.sg. This author is also affiliated with the Department of Computer Science, School of Computing, National University of Singapore. Web address: http://www.comp.nus.edu.sg/~nght
‡ 20 Science Park Drive, Singapore 118230. E-mail: lchungyo@dso.org.sg

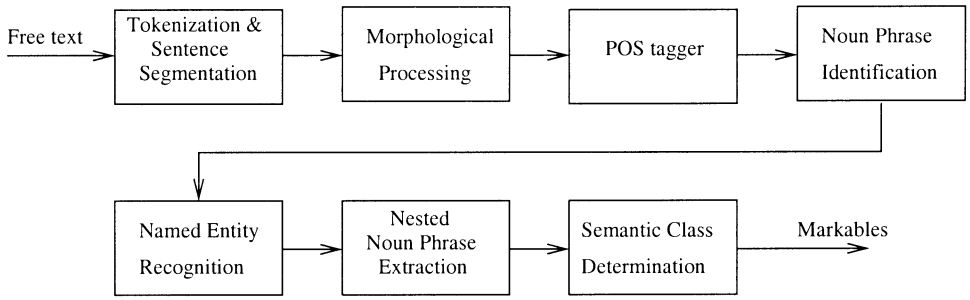

Figure 1
System architecture of natural language processing pipeline.

2. A Machine Learning Approach to Coreference Resolution

We adopt a corpus-based, machine learning approach to noun phrase coreference resolution. This approach requires a relatively small corpus of training documents that have been annotated with coreference chains of noun phrases. All possible markables in a training document are determined by a pipeline of language-processing modules, and training examples in the form of feature vectors are generated for appropriate pairs of markables. These training examples are then given to a learning algorithm to build a classifier. To determine the coreference chains in a new document, all markables are determined and potential pairs of coreferring markables are presented to the classifier, which decides whether the two markables actually corefer. We give the details of these steps in the following subsections.

2.1 Determination of Markables

A prerequisite for coreference resolution is to obtain most, if not all, of the possible markables in a raw input text. To determine the markables, a pipeline of natural language processing (NLP) modules is used, as shown in Figure 1. They consist of tokenization, sentence segmentation, morphological processing, part-of-speech tagging, noun phrase identification, named entity recognition, nested noun phrase extraction, and semantic class determination. As far as coreference resolution is concerned, the goal of these NLP modules is to determine the boundary of the markables, and to provide the necessary information about each markable for subsequent generation of features in the training examples.

Our part-of-speech tagger is a standard statistical tagger based on the Hidden Markov Model (HMM) (Church 1988). Similarly, we built a statistical HMM-based noun phrase identification module that determines the noun phrase boundaries solely based on the part-of-speech tags assigned to the words in a sentence. We also implemented a module that recognizes MUC-style named entities, that is, organization, person, location, date, time, money, and percent. Our named entity recognition module uses the HMM approach of Bikel, Schwartz, and Weischedel (1999), which learns from a tagged corpus of named entities. That is, our part-of-speech tagger, noun phrase identification module, and named entity recognition module are all based on HMMs and learn from corpora tagged with parts of speech, noun phrases, and named entities, respectively. Next, both the noun phrases determined by the noun phrase identification module and the named entities are merged in such a way that if the noun phrase overlaps with a named entity, the noun phrase boundaries will be adjusted to subsume the named entity.

The nested noun phrase extraction module subsequently accepts the noun phrases and determines the nested phrases for each noun phrase. The nested noun phrases are divided into two groups:

1. Nested noun phrases from possessive noun phrases. Consider two possessive noun phrases marked by the noun phrase module, *his long-range strategy* and *Eastern's parent*. The nested noun phrase for the first phrase is the pronoun *his*, while for the second one, it is the proper name *Eastern*.
2. Nested noun phrases that are modifier nouns (or prenominals). For example, the nested noun phrase for *wage reductions* is *wage*, and for *Union representatives*, it is *Union*.

Finally, the markables needed for coreference resolution are the union of the noun phrases, named entities, and nested noun phrases found. For markables without any named entity type, semantic class is determined by the semantic class determination module. More details regarding this module are given in the description of the semantic class agreement feature.

To achieve acceptable recall for coreference resolution, it is most critical that the eligible candidates for coreference be identified correctly in the first place. In order to test our system's effectiveness in determining the markables, we attempted to match the markables generated by our system against those appearing in the coreference chains annotated in 100 SGML documents, a subset of the training documents available in MUC-6. We found that our system is able to correctly identify about 85% of the noun phrases appearing in coreference chains in the 100 annotated SGML documents. Most of the unmatched noun phrases are of the following types:

1. Our system generated a head noun that is a subset of the noun phrase in the annotated corpus. For example, *Saudi Arabia, the cartel's biggest producer* was annotated as a markable, but our system generated only *Saudi Arabia*.
2. Our system extracted a sequence of words that cannot be considered as a markable.
3. Our system extracted markables that appear to be correct but do not match what was annotated. For example, our system identified *selective wage reductions*, but *wage reductions* was annotated instead.

2.2 Determination of Feature Vectors

To build a learning-based coreference engine, we need to devise a set of features that is useful in determining whether two markables corefer or not. In addition, these features must be generic enough to be used across different domains. Since the MUC-6 and MUC-7 tasks define coreference guidelines for all types of noun phrases and different types of noun phrases behave differently in terms of how they corefer, our features must be able to handle this and give different coreference decisions based on different types of noun phrases. In general, there must be some features that indicate the type of a noun phrase. Altogether, we have five features that indicate whether the markables are definite noun phrases, demonstrative noun phrases, pronouns, or proper names.

There are many important knowledge sources useful for coreference. We wanted to use those that are not too difficult to compute. One important factor is the distance

between the two markables. McEnery, Tanaka, and Botley (1997) have done a study on how distance affects coreference, particularly for pronouns. One of their conclusions is that the antecedents of pronouns do exhibit clear quantitative patterns of distribution. The distance feature has different effects on different noun phrases. For proper names, locality of the antecedents may not be so important. We include the distance feature so that the learning algorithm can best decide the distribution for different classes of noun phrases.

There are other features that are related to the gender, number, and semantic class of the two markables. Such knowledge sources are commonly used for the task of determining coreference.

Our feature vector consists of a total of 12 features described below, and is derived based on two extracted markables, i and j, where i is the potential antecedent and j is the anaphor. Information needed to derive the feature vectors is provided by the pipeline of language-processing modules prior to the coreference engine.

1. **Distance Feature (DIST):** Its possible values are $0, 1, 2, 3, \ldots$. This feature captures the distance between i and j. If i and j are in the same sentence, the value is 0; if they are one sentence apart, the value is 1; and so on.
2. **i-Pronoun Feature (I_PRONOUN):** Its possible values are true or false. If i is a pronoun, return true; else return false. Pronouns include reflexive pronouns (*himself, herself*), personal pronouns (*he, him, you*), and possessive pronouns (*hers, her*).
3. **j-Pronoun Feature (J_PRONOUN):** Its possible values are true or false. If j is a pronoun (as described above), then return true; else return false.
4. **String Match Feature (STR_MATCH):** Its possible values are true or false. If the string of i matches the string of j, return true; else return false. We first remove articles (*a, an, the*) and demonstrative pronouns (*this, these, that, those*) from the strings before performing the string comparison. Therefore, *the license* matches *this license*, *that computer* matches *computer*.
5. **Definite Noun Phrase Feature (DEF_NP):** Its possible values are true or false. In our definition, a definite noun phrase is a noun phrase that starts with the word *the*. For example, *the car* is a definite noun phrase. If j is a definite noun phrase, return true; else return false.
6. **Demonstrative Noun Phrase Feature (DEM_NP):** Its possible values are true or false. A demonstrative noun phrase is one that starts with the word *this, that, these*, or *those*. If j is a demonstrative noun phrase, then return true; else return false.
7. **Number Agreement Feature (NUMBER):** Its possible values are true or false. If i and j agree in number (i.e., they are both singular or both plural), the value is true; otherwise false. Pronouns such as *they* and *them* are plural, while *it, him*, and so on, are singular. The morphological root of a noun is used to determine whether it is singular or plural if the noun is not a pronoun.
8. **Semantic Class Agreement Feature (SEMCLASS):** Its possible values are true, false, or unknown. In our system, we defined the following semantic classes: "female," "male," "person," "organization," "location," "date," "time," "money," "percent," and "object." These semantic classes

are arranged in a simple ISA hierarchy. Each of the "female" and "male" semantic classes is a subclass of the semantic class "person," while each of the semantic classes "organization," "location," "date," "time," "money," and "percent" is a subclass of the semantic class "object." Each of these defined semantic classes is then mapped to a WordNet synset (Miller 1990). For example, "male" is mapped to the second sense of the noun *male* in WordNet, "location" is mapped to the first sense of the noun *location*, and so on.

The semantic class determination module assumes that the semantic class for every markable extracted is the first sense of the head noun of the markable. Since WordNet orders the senses of a noun by their frequency, this is equivalent to choosing the most frequent sense as the semantic class for each noun. If the selected semantic class of a markable is a subclass of one of our defined semantic classes *C*, then the semantic class of the markable is *C*; else its semantic class is "unknown."

The semantic classes of markables *i* and *j* are in agreement if one is the parent of the other (e.g., *chairman* with semantic class "person" and *Mr. Lim* with semantic class "male"), or they are the same (e.g., *Mr. Lim* and *he*, both of semantic class "male"). The value returned for such cases is true. If the semantic classes of *i* and *j* are not the same (e.g., *IBM* with semantic class "organization" and *Mr. Lim* with semantic class "male"), return false. If either semantic class is "unknown," then the head noun strings of both markables are compared. If they are the same, return true; else return unknown.

9. **Gender Agreement Feature (GENDER):** Its possible values are true, false, or unknown. The gender of a markable is determined in several ways. Designators and pronouns such as *Mr., Mrs., she*, and *he*, can determine the gender. For a markable that is a person's name, such as *Peter H. Diller*, the gender cannot be determined by the above method. In our system, the gender of such a markable can be determined if markables are found later in the document that refer to *Peter H. Diller* by using the designator form of the name, such as *Mr. Diller*. If the designator form of the name is not present, the system will look through its database of common human first names to determine the gender of that markable. The gender of a markable will be unknown for noun phrases such as *the president* and *chief executive officer*. The gender of other markables that are not "person" is determined by their semantic classes. Unknown semantic classes will have unknown gender while those that are objects will have "neutral" gender. If the gender of either markable *i* or *j* is unknown, then the gender agreement feature value is unknown; else if *i* and *j* agree in gender, then the feature value is true; otherwise its value is false.

10. **Both-Proper-Names Feature (PROPER_NAME):** Its possible values are true or false. A proper name is determined based on capitalization. Prepositions appearing in the name such as *of* and *and* need not be in uppercase. If *i* and *j* are both proper names, return true; else return false.

11. **Alias Feature (ALIAS):** Its possible values are true or false. If *i* is an alias of *j* or vice versa, return true; else return false. That is, this feature value is true if *i* and *j* are named entities (person, date, organization, etc.) that refer to the same entity. The alias module works differently depending

on the named entity type. For *i* and *j* that are dates (e.g., *01-08* and *Jan. 8*), by using string comparison, the day, month, and year values are extracted and compared. If they match, then *j* is an alias of *i*. For *i* and *j* that are "person," such as *Mr. Simpson* and *Bent Simpson*, the last words of the noun phrases are compared to determine whether one is an alias of the other. For organization names, the alias function also checks for acronym match such as *IBM* and *International Business Machines Corp.* In this case, the longer string is chosen to be the one that is converted into the acronym form. The first step is to remove all postmodifiers such as *Corp.* and *Ltd.* Then, the acronym function considers each word in turn, and if the first letter is capitalized, it is used to form the acronym. Two variations of the acronyms are produced: one with a period after each letter, and one without.

12. **Appositive Feature (APPOSITIVE):** Its possible values are true or false. If *j* is in apposition to *i*, return true; else return false. For example, the markable *the chairman of Microsoft Corp.* is in apposition to *Bill Gates* in the sentence *Bill Gates, the chairman of Microsoft Corp., ...* Our system determines whether *j* is a possible appositive construct by first checking for the existence of verbs and proper punctuation. Like the above example, most appositives do not have any verb; and an appositive is separated by a comma from the most immediate antecedent, *i*, to which it refers. Further, at least one of *i* and *j* must be a proper name. The MUC-6 and MUC-7 coreference task definitions are slightly different. In MUC-6, *j* needs to be a definite noun phrase to be an appositive, while both indefinite and definite noun phrases are acceptable in MUC-7.

As an example, Table 1 shows the feature vector associated with the antecedent *i*, *Frank Newman*, and the anaphor *j*, *vice chairman*, in the following sentence:

(1) Separately, Clinton transition officials said that *Frank Newman*, 50, *vice chairman* and chief financial officer of BankAmerica Corp., is expected to be nominated as assistant Treasury secretary for domestic finance.

Table 1
Feature vector of the markable pair (*i* = *Frank Newman*, *j* = *vice chairman*).

Feature	Value	Comments
DIST	0	*i* and *j* are in the same sentence
I_PRONOUN	–	*i* is not a pronoun
J_PRONOUN	–	*j* is not a pronoun
STR_MATCH	–	*i* and *j* do not match
DEF_NP	–	*j* is not a definite noun phrase
DEM_NP	–	*j* is not a demonstrative noun phrase
NUMBER	+	*i* and *j* are both singular
SEMCLASS	1	*i* and *j* are both persons (This feature has three values: false(0), true(1), unknown(2).)
GENDER	1	*i* and *j* are both males (This feature has three values: false(0), true(1), unknown(2).)
PROPER_NAME	–	Only *i* is a proper name
ALIAS	–	*j* is not an alias of *i*
APPOSITIVE	+	*j* is in apposition to *i*

Because of capitalization, markables in the headlines of MUC-6 and MUC-7 documents are always considered proper names even though some are not. Our system solves this inaccuracy by first preprocessing a headline to correct the capitalization before passing it into the pipeline of NLP modules. Only those markables in the headline that appear in the text body as proper names have their capitalization changed to match those found in the text body. All other headline markables are changed to lowercase.

2.3 Generating Training Examples

Consider a coreference chain A1 - A2 - A3 - A4 found in an annotated training document. Only pairs of noun phrases in the chain that are immediately adjacent (i.e., A1 - A2, A2 - A3, and A3 - A4) are used to generate the positive training examples. The first noun phrase in a pair is always considered the antecedent, while the second is the anaphor. On the other hand, negative training examples are extracted as follows. Between the two members of each antecedent-anaphor pair, there are other markables extracted by our language-processing modules that either are not found in any coreference chain or appear in other chains. Each of them is then paired with the anaphor to form a negative example. For example, if markables a, b, and B1 appear between A1 and A2, then the negative examples are a - A2, b - A2, and B1 - A2. Note that a and b do not appear in any coreference chain, while B1 appears in another coreference chain.

For an annotated noun phrase in a coreference chain in a training document, the same noun phrase must be identified as a markable by our pipeline of language-processing modules before this noun phrase can be used to form a feature vector for use as a training example. This is because the information necessary to derive a feature vector, such as semantic class and gender, is computed by the language-processing modules. If an annotated noun phrase is not identified as a markable, it will not contribute any training example. To see more clearly how training examples are generated, consider the following four sentences:

- Sentence 1
  1. (Eastern Air)$_{a1}$ Proposes (Date For Talks on ((Pay)$_{c1}$-Cut)$_{d1}$ Plan)$_{b1}$
  2. (Eastern Air)$_1$ Proposes (Date)$_2$ For (Talks)$_3$ on (Pay-Cut Plan)$_4$
- Sentence 2
  1. (Eastern Airlines)$_{a2}$ executives notified (union)$_{e1}$ leaders that the carrier wishes to discuss selective ((wage)$_{c2}$ reductions)$_{d2}$ on (Feb. 3)$_{b2}$.
  2. ((Eastern Airlines)$_5$ executives)$_6$ notified ((union)$_7$ leaders)$_8$ that (the carrier)$_9$ wishes to discuss (selective (wage)$_{10}$ reductions)$_{11}$ on (Feb. 3)$_{12}$.
- Sentence 3
  1. ((Union)$_{e2}$ representatives who could be reached)$_{f1}$ said (they)$_{f2}$ hadn't decided whether (they)$_{f3}$ would respond.
  2. ((Union)$_{13}$ representatives)$_{14}$ who could be reached said (they)$_{15}$ hadn't decided whether (they)$_{16}$ would respond.

- Sentence 4

1. By proposing (a meeting date)$_{b3}$, (Eastern)$_{a3}$ moved one step closer toward reopening current high-cost contract agreements with ((its)$_{a4}$ unions)$_{e3}$.
2. By proposing (a meeting date)$_{17}$, (Eastern)$_{18}$ moved (one step)$_{19}$ closer toward reopening (current high-cost contract agreements)$_{20}$ with ((its)$_{21}$ unions)$_{22}$.

Each sentence is shown twice with different noun phrase boundaries. Sentences labeled (1) are obtained directly from part of the training document. The letters in the subscripts uniquely identify the coreference chains, while the numbers identify the noun phrases. Noun phrases in sentences labeled (2) are extracted by our language-processing modules and are also uniquely identified by numeric subscripts.

Let's consider chain *e*, which is about the union. There are three noun phrases that corefer, and our system managed to extract the boundaries that correspond to all of them: (*union*)$_7$ matches with (*union*)$_{e1}$, (*union*)$_{13}$ with (*union*)$_{e2}$, and (*its unions*)$_{22}$ with (*its unions*)$_{e3}$. There are two positive training examples formed by ((*union*)$_{13}$, (*its unions*)$_{22}$) and ((*union*)$_7$, (*union*)$_{13}$). Noun phrases between (*union*)$_7$ and (*union*)$_{13}$ that do not corefer with (*union*)$_{13}$ are used to form the negative examples. The negative examples are ((*the carrier*)$_9$, (*union*)$_{13}$), ((*wage*)$_{10}$, (*union*)$_{13}$), ((*selective wage reductions*)$_{11}$, (*union*)$_{13}$), and ((*Feb. 3*)$_{12}$, (*union*)$_{13}$). Negative examples can also be found similarly between ((*union*)$_{13}$, (*its unions*)$_{22}$).

As another example, neither noun phrase in chain *d*, (*Pay-Cut*)$_{d1}$ and (*wage reductions*)$_{d2}$, matches with any machine-extracted noun phrase boundaries. In this case, no positive or negative example is formed for noun phrases in chain *d*.

2.4 Building a Classifier

The next step is to use a machine learning algorithm to learn a classifier based on the feature vectors generated from the training documents. The learning algorithm used in our coreference engine is C5, which is an updated version of C4.5 (Quinlan 1993). C5 is a commonly used decision tree learning algorithm and thus it may be considered as a baseline method against which other learning algorithms can be compared.

2.5 Generating Coreference Chains for Test Documents

Before determining the coreference chains for a test document, all possible markables need to be extracted from the document. Every markable is a possible anaphor, and every markable before the anaphor in document order is a possible antecedent of the anaphor, except when the anaphor is nested. If the anaphor is a child or nested markable, then its possible antecedents must not be any markable with the same root markable as the current anaphor. However, the possible antecedents can be other root markables and their children that are before the anaphor in document order. For example, consider the two root markables, *Mr. Tom's daughter* and *His daughter's eyes*, appearing in that order in a test document. The possible antecedents of *His* cannot be *His daughter* or *His daughter's eyes*, but can be *Mr. Tom* or *Mr. Tom's daughter*.

The coreference resolution algorithm considers every markable *j* starting from the second markable in the document to be a potential candidate as an anaphor. For each *j*, the algorithm considers every markable *i* before *j* as a potential antecedent. For each pair *i* and *j*, a feature vector is generated and given to the decision tree classifier. A coreferring antecedent is found if the classifier returns true. The algorithm starts from the immediately preceding markable and proceeds backward in the reverse order of

the markables in the document until there is no remaining markable to test or an antecedent is found.

As an example, consider the following text with markables already detected by the NLP modules:

(2) **(Ms. Washington)**$_{73}$'s candidacy is being championed by (several powerful lawmakers)$_{74}$ including ((**her**)$_{76}$ boss)$_{75}$, (Chairman John Dingell)$_{77}$ (D., (Mich.)$_{78}$) of (the House Energy and Commerce Committee)$_{79}$. **(She)**$_{80}$ currently is (a counsel)$_{81}$ to (the committee)$_{82}$. **(Ms. Washington)**$_{83}$ and (Mr. Dingell)$_{84}$ have been considered (allies)$_{85}$ of (the (securities)$_{87}$ exchanges)$_{86}$, while (banks)$_{88}$ and ((futures)$_{90}$ exchanges)$_{89}$ have often fought with (them)$_{91}$.

We will consider how the boldfaced chains are detected. Table 2 shows the pairs of markables tested for coreference to form the chain for *Ms. Washington-her-She-Ms. Washington*. When the system considers the anaphor, (*her*)$_{76}$, all preceding phrases, except (*her boss*)$_{75}$, are tested to see whether they corefer with it. (*her boss*)$_{75}$ is not tested because (*her*)$_{76}$ is its nested noun phrase. Finally, the decision tree determines that the noun phrase (*Ms. Washington*)$_{73}$ corefers with (*her*)$_{76}$. In Table 2, we only show the system considering the three anaphors (*her*)$_{76}$, (*She*)$_{80}$, and (*Ms. Washington*)$_{83}$, in that order.

Table 2
Pairs of markables that are tested in forming the coreference chain *Ms. Washington-her-She-Ms. Washington*. The feature vector format: DIST, SEMCLASS, NUMBER, GENDER, PROPER_NAME, ALIAS, J_PRONOUN, DEF_NP, DEM_NP, STR_MATCH, APPOSITIVE, I_PRONOUN.

Antecedent	Anaphor	Feature Vector	Corefers?
(several powerful lawmakers)$_{74}$	(her)$_{76}$	0,1,−,2,−,−,+,−,−,−,−,−	No
(Ms. Washington)$_{73}$	(her)$_{76}$	0,1,+,1,−,−,+,−,−,−,−,−	Yes
(the House Energy and Commerce Committee)$_{79}$	(She)$_{80}$	1,0,+,0,−,−,+,−,−,−,−,−	No
(Mich.)$_{78}$	(She)$_{80}$	2,0,+,0,−,−,+,−,−,−,−,−	No
(Chairman John Dingell)$_{77}$	(She)$_{80}$	3,1,+,0,−,−,+,−,−,−,−,−	No
(her)$_{76}$	(She)$_{80}$	3,1,+,1,−,−,+,−,−,−,−,+	Yes
(the committee)$_{82}$	(Ms. Washington)$_{83}$	1,0,+,0,−,−,−,−,−,−,−,−	No
(a counsel)$_{81}$	(Ms. Washington)$_{83}$	1,1,+,2,−,−,−,−,−,−,−,−	No
(She)$_{80}$	(Ms. Washington)$_{83}$	1,1,+,1,−,−,−,−,−,−,−,+	No
(the House Energy and Commerce Committee)$_{79}$	(Ms. Washington)$_{83}$	2,0,+,0,+,−,−,−,−,−,−,−	No
(Mich.)$_{78}$	(Ms. Washington)$_{83}$	3,0,+,0,+,−,−,−,−,−,−,−	No
(Chairman John Dingell)$_{77}$	(Ms. Washington)$_{83}$	4,1,+,0,+,−,−,−,−,−,−,−	No
(her)$_{76}$	(Ms. Washington)$_{83}$	4,1,+,1,−,−,−,−,−,−,−,+	No
(her boss)$_{75}$	(Ms. Washington)$_{83}$	4,1,−,0,−,−,−,−,−,−,−,−	No
(several powerful lawmakers)$_{74}$	(Ms. Washington)$_{83}$	4,1,−,2,−,−,−,−,−,−,−,−	No
(Ms. Washington)$_{73}$	(Ms. Washington)$_{83}$	4,1,+,1,+,+,−,−,−,+,−,−	Yes

We use the same method to generate coreference chains for both MUC-6 and MUC-7, except for the following. For MUC-7, because of slight changes in the coreference task definition, we include a filtering module to remove certain coreference chains. The task definition states that a coreference chain must contain at least one element that is a head noun or a name; that is, a chain containing only prenominal modifiers is removed by the filtering module.

3. Evaluation

In order to evaluate the performance of our learning approach to coreference resolution on common data sets, we utilized the annotated corpora and scoring programs from MUC-6 and MUC-7, which assembled a set of newswire documents annotated with coreference chains. Although we did not participate in either MUC-6 or MUC-7, we were able to obtain the training and test corpora for both years from the MUC organizers for research purposes.[1] To our knowledge, these are the only publicly available annotated corpora for coreference resolution.

For MUC-6, 30 dry-run documents annotated with coreference information were used as the training documents for our coreference engine. There are also 30 annotated training documents from MUC-7. The total size of the 30 training documents is close to 12,400 words for MUC-6 and 19,000 words for MUC-7. There are altogether 20,910 (48,872) training examples used for MUC-6 (MUC-7), of which only 6.5% (4.4%) are positive examples in MUC-6 (MUC-7).[2]

After training a separate classifier for each year, we tested the performance of each classifier on its corresponding test corpus. For MUC-6, the C5 pruning confidence is set at 20% and the minimum number of instances per leaf node is set at 5. For MUC-7, the pruning confidence is 60% and the minimum number of instances is 2. The parameters are determined by performing 10-fold cross-validation on the whole training set for each MUC year. The possible pruning confidence values that we tried are 10%, 20%, 40%, 60%, 80%, and 100%, and for minimum instances, we tried 2, 5, 10, 15, and 20. Thus, a total of 30 (6×5) cross-validation runs were executed.

One advantage of using a decision tree learning algorithm is that the resulting decision tree classifier can be interpreted by humans. The decision tree generated for MUC-6, shown in Figure 2, seems to encapsulate a reasonable rule of thumb that matches our intuitive linguistic notion of when two noun phrases can corefer. It is also interesting to note that only 8 out of the 12 available features in the training examples are actually used in the final decision tree built.

MUC-6 has a standard set of 30 test documents, which is used by all systems that participated in the evaluation. Similarly, MUC-7 has a test corpus of 20 documents. We compared our system's MUC-6 and MUC-7 performance with that of the systems that took part in MUC-6 and MUC-7, respectively. When the coreference engine is given new test documents, its output is in the form of SGML files with the coreference chains properly annotated according to the guidelines.[3] We then used the scoring programs

1 See http://www.itl.nist.gov/iad/894.02/related_projects/muc/index.html for details on obtaining the corpora.

2 Our system runs on a Pentium III 550MHz PC. It took less than 5 minutes to generate the training examples from the training documents for MUC-6, and about 7 minutes for MUC-7. The training time for the C5 algorithm to generate a decision tree from all the training examples was less than 3 seconds for both MUC years.

3 The time taken to generate the coreference chains for the 30 MUC-6 test documents of close to 13,400 words was less than 3 minutes, while it took less than 2 minutes for the 20 MUC-7 test documents of about 10,000 words.

```
STR_MATCH = +: +
STR_MATCH = -:
:...J_PRONOUN = -:
    :...APPOSITIVE = +: +
    :   APPOSITIVE = -:
    :   :...ALIAS = +: +
    :       ALIAS = -: -
    J_PRONOUN = +:
    :...GENDER = 0: -
        GENDER = 2: -
        GENDER = 1:
        :...I_PRONOUN = +: +
            I_PRONOUN = -:
            :...DIST > 0: -
                DIST <= 0:
                :...NUMBER = +: +
                    NUMBER = -: -
```

Figure 2
The decision tree classifier learned for MUC-6.

for the respective years to generate the recall and precision scores for our coreference engine.

Our coreference engine achieves a recall of 58.6% and a precision of 67.3%, yielding a balanced F-measure of 62.6% for MUC-6. For MUC-7, the recall is 56.1%, the precision is 65.5%, and the balanced F-measure is 60.4%.[4] We plotted the scores of our coreference engine (square-shaped) against the official test scores of the other systems (cross-shaped) in Figure 3 and Figure 4.

We also plotted the learning curves of our coreference engine in Figure 5 and Figure 6, showing its accuracy averaged over three random trials when trained on 1, 2, 3, 4, 5, 10, 15, 20, 25, and 30 training documents. The learning curves indicate that our coreference engine achieves its peak performance with about 25 training documents, or about 11,000 to 17,000 words of training documents. This number of training documents would generate tens of thousands of training examples, sufficient for the decision tree learning algorithm to learn a good classifier. At higher numbers of training documents, our system seems to start overfitting the training data. For example, on MUC-7 data, training on the full set of 30 training documents results in a more complex decision tree.

Our system's scores are in the upper region of the MUC-6 and MUC-7 systems. We performed a simple one-tailed, paired sample t-test at significance level $p = 0.05$ to determine whether the difference between our system's F-measure score and each of the other systems' F-measure score on the test documents is statistically significant.[5] We found that at the 95% significance level ($p = 0.05$), our system performed better than three MUC-6 systems, and as well as the rest of the MUC-6 systems. Using the

4 Note that MUC-6 did not use balanced F-measure as the official evaluation measure, but MUC-7 did.
5 Though the McNemar test is shown to have low Type I error compared with the paired t-test (Dietterich 1998), we did not carry out this test in the context of coreference. This is because an example instance defines a coreference link between two noun phrases, and since this link is transitive in nature, it is unclear how the number of links misclassified by System A but not by System B and vice versa can be obtained to execute the McNemar test.

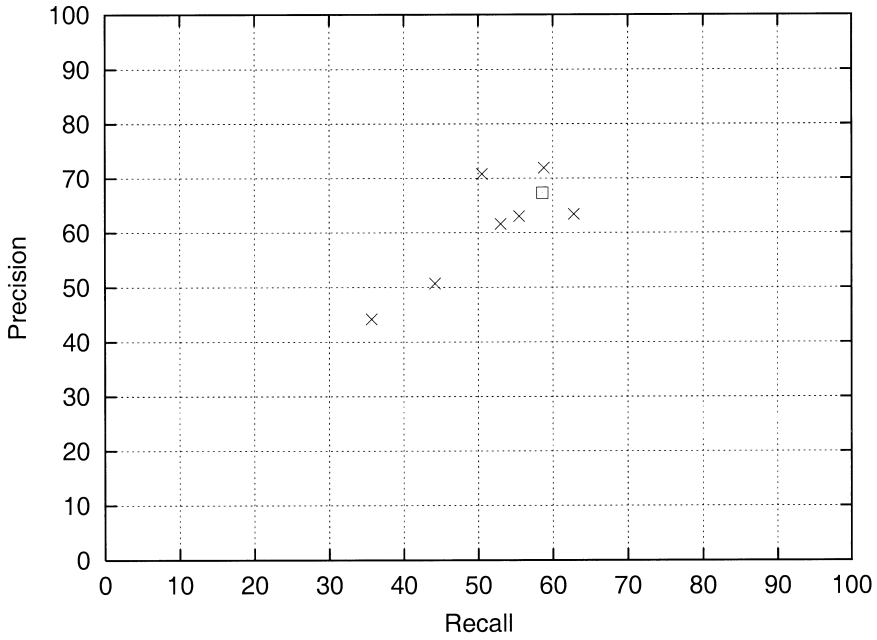

Figure 3
Coreference scores of MUC-6 systems and our system.

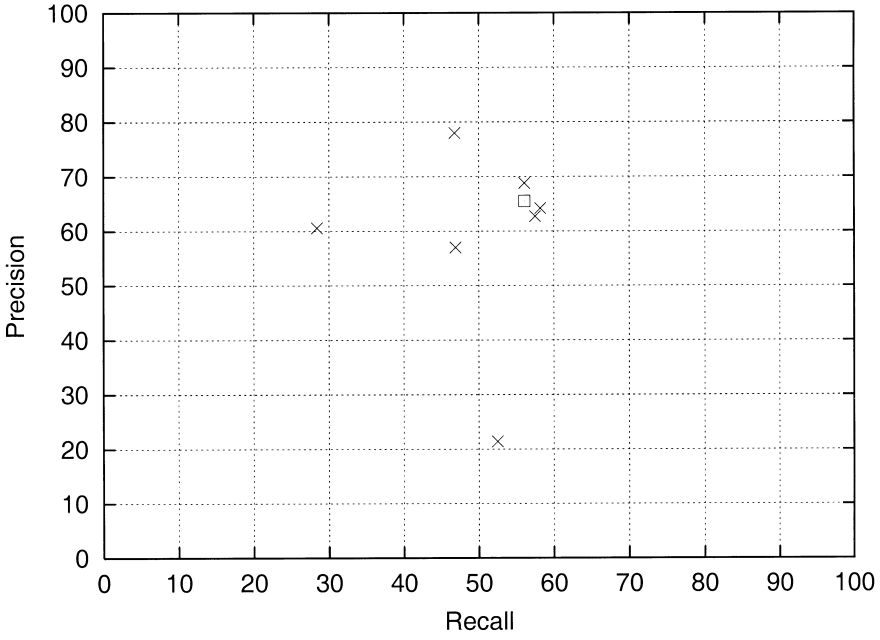

Figure 4
Coreference scores of MUC-7 systems and our system.

same significance level, our system performed better than four MUC-7 systems, and as well as the rest of the MUC-7 systems. Our result is encouraging since it indicates that a learning approach using relatively shallow features can achieve scores comparable to those of systems built using nonlearning approaches.

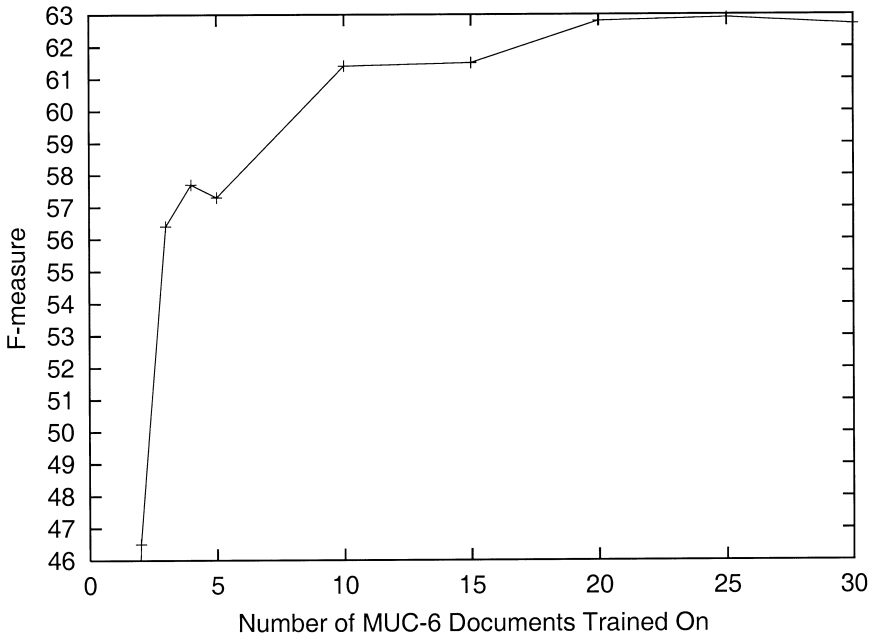

Figure 5
Learning curve of coreference resolution accuracy for MUC-6.

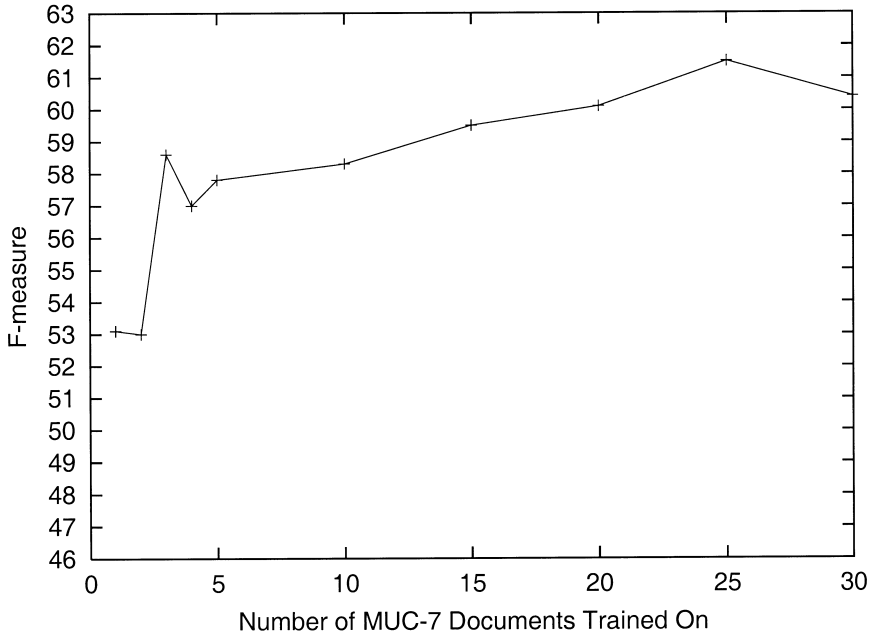

Figure 6
Learning curve of coreference resolution accuracy for MUC-7.

It should be noted that the accuracy of our coreference resolution engine depends to a large extent on the performance of the NLP modules that are executed before the coreference engine. Our current learning-based, HMM named entity recognition module is trained on 318 documents (a disjoint set from both the MUC-6 and MUC-7

test documents) tagged with named entities, and its score on the MUC-6 named entity task for the 30 formal test documents is only 88.9%, which is not considered very high by MUC-6 standards. For example, our named entity recognizer could not identify the two named entities *USAir* and *Piedmont* in the expression *USAir and Piedmont* but instead treat them as one single named entity. Our part-of-speech tagger achieves 96% accuracy, while the accuracy of noun phrase identification is above 90%.

4. The Contribution of the Features

One factor that affects the performance of a machine learning approach is the set of features used. It is interesting to find out how useful each of our 12 features is in the MUC-6 and MUC-7 coreference tasks. One way to do this is to train and test using just one feature at a time. Table 3 and Table 4 show the results of the experiment. For both MUC-6 and MUC-7, the 3 features that give nonzero recall and precision are ALIAS, STR_MATCH, and APPOSITIVE. The 12 features can be divided into unary and binary

Table 3
MUC-6 results of complete and baseline systems to study the contribution of the features.

System ID	Recall	Prec	F	Remarks
Complete systems				
DSO	58.6	67.3	62.6	Our system
DSO_TRG	52.6	67.6	59.2	Our system using RESOLVE's method of generating positive and negative examples
RESOLVE	44.2	50.7	47.2	The RESOLVE coreference system at the University of Massachusetts
Baseline systems using just one feature				
DIST	0.0	0.0	0.0	Only "distance" feature is used
SEMCLASS	0.0	0.0	0.0	Only "semantic class agreement"
NUMBER	0.0	0.0	0.0	Only "number agreement"
GENDER	0.0	0.0	0.0	Only "gender agreement"
PROPER_NAME	0.0	0.0	0.0	Only "both proper names"
ALIAS	24.5	88.7	38.4	Only "alias"
J_PRONOUN	0.0	0.0	0.0	Only "j-pronoun"
DEF_NP	0.0	0.0	0.0	Only "definite noun phrase"
DEM_NP	0.0	0.0	0.0	Only "demonstrative noun phrase"
STR_MATCH	45.7	65.6	53.9	Only "string match"
APPOSITIVE	3.9	57.7	7.3	Only "appositive"
I_PRONOUN	0.0	0.0	0.0	Only "i-pronoun"
Other baseline systems				
ALIAS_STR	51.5	66.4	58.0	Only the "alias" and "string match" features are used
ALIAS_STR_APPOS	55.2	66.4	60.3	Only the "alias," "string match," and "appositive" features are used
ONE_CHAIN	89.9	31.8	47.0	All markables form one chain
ONE_WRD	55.4	36.6	44.1	Markables corefer if there is at least one common word
HD_WRD	56.4	50.4	53.2	Markables corefer if their head words are the same

Table 4
MUC-7 results of complete and baseline systems to study the contribution of the features.

System ID	Recall	Prec	F	Remarks
Complete systems				
DSO	56.1	65.5	60.4	Our system
DSO_TRG	53.3	69.7	60.4	Our system using RESOLVE's method of generating positive and negative examples
Baseline systems using just one feature				
DIST	0.0	0.0	0.0	Only "distance" feature is used
SEMCLASS	0.0	0.0	0.0	Only "semantic class agreement"
NUMBER	0.0	0.0	0.0	Only "number agreement"
GENDER	0.0	0.0	0.0	Only "gender agreement"
PROPER_NAME	0.0	0.0	0.0	Only "both proper names"
ALIAS	25.6	81.1	38.9	Only "alias"
J_PRONOUN	0.0	0.0	0.0	Only "j-pronoun"
DEF_NP	0.0	0.0	0.0	Only "definite noun phrase"
DEM_NP	0.0	0.0	0.0	Only "demonstrative noun phrase"
STR_MATCH	43.8	71.4	54.3	Only "string match"
APPOSITIVE	2.4	60.0	4.6	Only "appositive"
I_PRONOUN	0.0	0.0	0.0	Only "i-pronoun"
Other baseline systems				
ALIAS_STR	49.4	70.4	58.1	Only the "alias" and "string match" features are used
ALIAS_STR_APPOS	51.6	69.9	59.4	Only the "alias," "string match," and "appositive" features are used
ONE_CHAIN	87.5	30.5	45.2	All markables form one chain
ONE_WRD	55.9	38.7	45.7	Markables corefer if there is at least one common word
HD_WRD	55.2	55.6	55.4	Markables corefer if their head words are the same

features. The unary features are I_PRONOUN, J_PRONOUN, DEF_NP, and DEM_NP, while the rest are binary in nature. All the unary features score an F-measure of 0. The binary features with 0 F-measure are DIST, PROPER_NAME, GENDER, SEMCLASS, and NUMBER.

The ALIAS, APPOSITIVE, and STR_MATCH features give nonzero F-measure. All these features give rather high precision scores ($>$ 80% for ALIAS, $>$ 65% for STR_MATCH, and $>$ 57% for APPOSITIVE). Since these features are highly informative, we were curious to see how much they contribute to our MUC-6 and MUC-7 results of 62.6% and 60.4%, respectively. Systems ALIAS_STR and ALIAS_STR_APPOS in Table 3 and Table 4 show the results of the experiment. In terms of absolute F-measure, the difference between using these three features and using all features is 2.3% for MUC-6 and 1% for MUC-7; in other words, the other nine features contribute just 2.3% and 1% more for each of the MUC years. These nine features will be the first ones to be considered for pruning away by the C5 algorithm. For example, four features, namely, SEMCLASS, PROPER_NAME, DEF_NP, and DEM_NP, are not used in the MUC-6 tree shown in Figure 2. Figure 7 shows the distribution of the test cases over the five positive leaf nodes of the MUC-6 tree. For example, about 66.3% of all

```
STR_MATCH = +: +  944 (66.3%)

STR_MATCH = -:
:...J_PRONOUN = -:
    :...APPOSITIVE = +: +  111 (7.8%)

STR_MATCH = -:
:...J_PRONOUN = -:
    :...APPOSITIVE = -:
        :...ALIAS = +: +  163 (11.5%)

STR_MATCH = -:
:...J_PRONOUN = +:
    :...GENDER = 1:
        :...I_PRONOUN = +: +  77 (5.4%)

STR_MATCH = -:
:...J_PRONOUN = +:
    :...GENDER = 1:
        :...I_PRONOUN = -:
            :...DIST <= 0:
                :...NUMBER = +: +  128 (9.0%)
```

Figure 7
Distribution of test examples from the 30 MUC-6 test documents for positive leaf nodes of the MUC-6 tree.

the test examples that are classified positive go to the "If STR_MATCH" branch of the tree.

Other baseline systems that are used are ONE_CHAIN, ONE_WRD, and HD_WRD (Cardie and Wagstaff 1999). For ONE_CHAIN, all markables formed one chain. In ONE_WRD, markables corefer if there is at least one common word. In HD_WRD, markables corefer if their head words are the same. The purpose of ONE_CHAIN is to determine the maximum recall our system is capable of. The recall level here indirectly measures how effective the noun phrase identification module is. Both ONE_WRD and HD_WRD are less stringent variations of STR_MATCH. The performance of ONE_WRD is the worst. HD_WRD offers better recall compared to STR_MATCH, but poorer precision. However, its F-measure is comparable to that of STR_MATCH.

The score of the coreference system at the University of Massachusetts (RESOLVE), which uses C4.5 for coreference resolution, is shown in Table 3. RESOLVE is shown because among the MUC-6 systems, it is the only machine learning–based system that we can directly compare to. The other MUC-6 systems were not based on a learning approach. Also, none of the systems in MUC-7 adopted a learning approach to coreference resolution (Chinchor 1998).

RESOLVE's score is not high compared to scores attained by the rest of the MUC-6 systems. In particular, the system's recall is relatively low. Our system's score is higher than that of RESOLVE, and the difference is statistically significant. The RESOLVE system is described in three papers: McCarthy and Lehnert (1995), Fisher et al. (1995), and McCarthy (1996). As explained in McCarthy (1996), the reason for this low recall is that RESOLVE takes only the "relevant entities" and "relevant references" as input, where the relevant entities and relevant references are restricted to "person"

and "organization." In addition, because of limitations of the noun phrase detection module, nested phrases are not extracted and therefore do not take part in coreference. Nested phrases can include prenominal modifiers, possessive pronouns, and so forth. Therefore, the number of candidate markables to be used for coreference is small.

On the other hand, the markables extracted by our system include nested noun phrases, MUC-style named entity types (money, percent, date, etc.), and other types not defined by MUC. These markables will take part in coreference. About 3,600 top-level markables are extracted from the 30 MUC-6 test documents by our system. As detected by our NLP modules, only about 35% of these 3,600 phrases are "person" and "organization" entities and references. Concentrating on just these types has thus affected the overall recall of the RESOLVE system.

RESOLVE's way of generating training examples also differs from our system's: instances are created for all possible pairings of "relevant entities" and "relevant references," instead of our system's method of stopping at the first coreferential noun phrase when traversing back from the anaphor under consideration. We implemented RESOLVE's way of generating training examples, and the results (DSO-TRG) are reported in Table 3 and Table 4. For MUC-7, there is no drop in F-measure; for MUC-6, the F-measure dropped slightly.

RESOLVE makes use of 39 features, considerably more than our system's 12 features. RESOLVE's feature set includes the two highly informative features, ALIAS and STR_MATCH. RESOLVE does not use the APPOSITIVE feature.

5. Error Analysis

In order to determine the major classes of errors made by our system, we randomly chose five test documents from MUC-6 and determined the coreference links that were either missing (false negatives) or spurious (false positives) in these sample documents. Missing links result in recall errors; spurious links result in precision errors.

Breakdowns of the number of spurious and missing links are shown in Table 5 and Table 6, respectively. The following two subsections describe the errors in more detail.

5.1 Errors Causing Spurious Links

This section describes the five major types of errors summarized in Table 5 in more detail.

5.1.1 Prenominal Modifier String Match. This class of errors occurs when some strings of the prenominal modifiers of two markables match by surface string comparison and thus, by the C5 decision tree in Figure 2, the markables are treated as coreferring. How-

Table 5
The types and frequencies of errors that affect precision.

Types of Errors Causing Spurious Links	Frequency	%
Prenominal modifier string match	16	42.1%
Strings match but noun phrases refer to different entities	11	28.9%
Errors in noun phrase identification	4	10.5%
Errors in apposition determination	5	13.2%
Errors in alias determination	2	5.3%

Table 6
The types and frequencies of errors that affect recall.

Types of Errors Causing Missing Links	Frequency	%
Inadequacy of current surface features	38	63.3%
Errors in noun phrase identification	7	11.7%
Errors in semantic class determination	7	11.7%
Errors in part-of-speech assignment	5	8.3%
Errors in apposition determination	2	3.3%
Errors in tokenization	1	1.7%

ever, the entire markable actually does not corefer. The nested noun phrase extraction module is responsible for obtaining the possible prenominal modifiers from a noun phrase.

In (3), the noun phrase extraction module mistakenly extracted $(vice)_1$ and $(vice)_2$, which are not prenominal modifiers. Because of string match, $(vice)_1$ and $(vice)_2$ incorrectly corefer. In (4), $(undersecretary)_2$ was correctly extracted as a prenominal modifier, but incorrectly corefers with $(undersecretary)_1$ by string match.

(3) David J. Bronczek, **(vice)**$_1$ president and general manager of Federal Express Canada Ltd., was named senior **(vice)**$_2$ president, Europe, Africa and Mediterranean, at this air-express concern.

(4) Tarnoff, a former Carter administration official and president of the Council on Foreign Relations, is expected to be named **(undersecretary)**$_1$ for political affairs. . . . Former Sen. Tim Wirth is expected to get a newly created **(undersecretary)**$_2$ post for global affairs, which would include refugees, drugs and environmental issues.

5.1.2 Strings Match but Noun Phrases Refer to Different Entities. This error occurs when the surface strings of two markables match and thus, by the C5 decision tree in Figure 2, they are treated as coreferring. However, they actually refer to different entities and should not corefer. In (5), $(the\ committee)_1$ actually refers to the entity *the House Energy and Commerce Committee*, and $(the\ committee)_2$ refers to *the Senate Finance Committee*; therefore, they should not corefer. In (6), the two instances of *chief executive officer* refer to two different persons, namely, *Allan Laufgraben* and *Milton Petrie*, and, again, should not corefer.

(5) Ms. Washington's candidacy is being championed by several powerful lawmakers including her boss, Chairman John Dingell (D., Mich.) of the House Energy and Commerce Committee. She currently is a counsel to **(the committee)**$_1$. . . . Mr. Bentsen, who headed the Senate Finance Committee for the past six years, also is expected to nominate Samuel Sessions, **(the committee)**$_2$'s chief tax counsel, to one of the top tax jobs at Treasury.

(6) Directors also approved the election of Allan Laufgraben, 54 years old, as president and **(chief executive officer)**$_1$ and Peter A. Left, 43, as chief operating officer. Milton Petrie, 90-year-old chairman, president and **(chief executive officer)**$_2$ since the company was founded in 1932, will continue as chairman.

5.1.3 Errors in Noun Phrase Identification. This class of errors is caused by mistakes made by the noun phrase identification module. In (7), *May* and *June* are incorrectly grouped together by the noun phrase identification module as one noun phrase, that is, *May, June*. This markable then incorrectly causes the APPOSITIVE feature to be true, which results in classifying the pair as coreferential. In fact, (*the first week of July*)$_2$ should not be in apposition to (*May, June*)$_1$. However, we classified this error as a noun phrase identification error because it is the first module that causes the error. In (8), the noun phrase module extracted *Metaphor Inc.* instead of *Metaphor Inc. unit*. This causes (*it*)$_2$ to refer to *Metaphor Inc.* instead of *Metaphor Inc. unit*.

(7) The women's apparel specialty retailer said sales at stores open more than one year, a key barometer of a retail concern's strength, declined 2.5% in **(May, June)**$_1$ and **(the first week of July)**$_2$.

(8) ... International Business Machines Corp.'s **(Metaphor Inc.)**$_1$ unit said **(it)**$_2$ will shed 80 employees ...

5.1.4 Errors in Apposition Determination. This class of errors occurs when the anaphor is incorrectly treated as being in apposition to the antecedent and therefore causes the noun phrases to corefer. The precision scores obtained when using the APPOSITIVE feature alone are shown in Table 3 and Table 4, which suggest that the module can be improved further. Examples where apposition determination is incorrect are shown in (9) and (10).

(9) Clinton officials are said to be deciding between recently retired Rep. Matthew McHugh (D., **(N.Y.)**$_1$) and **(environmental activist)**$_2$ and transition official Gus Speth for the director of the Agency for International Development.

(10) Metaphor, a software subsidiary that IBM purchased in 1991, also named **(Chris Grejtak)**$_1$, **(43 years old)**$_2$, currently a senior vice president, president and chief executive officer.

5.1.5 Errors in Alias Determination. This class of errors occurs when the anaphor is incorrectly treated as an alias of the antecedent, thus causing the noun phrase pair to corefer. In (11), the two phrases (*House*)$_1$ and (*the House Energy and Commerce Committee*)$_2$ corefer because the ALIAS feature is incorrectly determined to be true.

(11) Consuela Washington, a longtime **(House)**$_1$ staffer and an expert in securities laws, is a leading candidate to be chairwoman of the Securities and Exchange Commission in the Clinton administration.... Ms. Washington's candidacy is being championed by several powerful lawmakers including her boss, Chairman John Dingell (D., Mich.) of **(the House Energy and Commerce Committee)**$_2$.

5.2 Errors Causing Missing Links

This subsection describes the six major classes of errors summarized in Table 6 in more detail.

5.2.1 Inadequacy of Current Surface Features. This class of errors is due to the inadequacy of the current surface features because they do not have information about

other words (such as the connecting conjunctions, prepositions, or verbs) and other knowledge sources that may provide important clues for coreference. As a result, the set of shallow features we used is unable to correctly classify the noun phrases in the examples below as coreferring.

Example (12) illustrates why resolving (*them*)$_2$ is difficult. (*allies*)$_1$, *securities exchanges*, *banks*, and *futures exchanges* are all possible antecedents of (*them*)$_2$, and the feature set must include more information to be able to pick the correct one. The conjunction *and* in (13) and *was named* in (16) are important cues to determine coreference. In addition, it may also be possible to capture noun phrases in predicate constructions like (17), where (*Mr. Gleason*)$_1$ is the subject and (*president*)$_2$ is the object.

(12) Ms. Washington and Mr. Dingell have been considered **(allies)**$_1$ of the securities exchanges, while banks and futures exchanges have often fought with **(them)**$_2$.

(13) Separately, Clinton transition officials said that Frank Newman, 50, **(vice chairman)**$_1$ and **(chief financial officer)**$_2$ of BankAmerica Corp., is expected to be nominated as assistant Treasury secretary for domestic finance.

(14) Separately, **(Clinton transition officials)**$_1$ said that Frank Newman, 50, vice chairman and chief financial officer of BankAmerica Corp., is expected to be nominated as assistant Treasury secretary for domestic finance. . . . As early as today, **(the Clinton camp)**$_2$ is expected to name five undersecretaries of state and several assistant secretaries.

(15) **(Metro-Goldwyn-Mayer Inc.)**$_1$ said it named Larry Gleason president of world-wide theatrical distribution of **(the movie studio)**$_2$'s distribution unit.

(16) . . . **(general manager)**$_1$ of Federal Express Canada Ltd., was named **(senior vice president)**$_2$, Europe, Africa and Mediterranean . . .

(17) **(Mr. Gleason)**$_1$, 55 years old, was **(president)**$_2$ of theatrical exhibition for Paramount Communications Inc., in charge of the company's 1,300 movie screens in 12 countries.

5.2.2 Errors in Noun Phrase Identification. This class of errors was described in Section 5.1.3. The noun phrase identification module may extract noun phrases that do not match the phrases in the coreference chain, therefore causing missing links and recall error.

5.2.3 Errors in Semantic Class Determination. These errors are caused by the wrong assignment of semantic classes to words. For example, (*Metaphor*)$_1$ should be assigned "organization" but it is assigned "unknown" in (18), and (*second-quarter*)$_2$ should be assigned "date" instead of "unknown" in (19). However, correcting these classes will still not cause the noun phrases in the examples to corefer. This is because the values of the SEMCLASS feature in the training examples are extremely noisy, a situation caused largely by our semantic class determination module. In many of the negative training examples, although the noun phrases are assigned the same semantic classes, these assignments do not seem to be correct. Some examples are (*four-year, NBC*), (*The union, Ford Motor Co.*), and (*base-wage, job-security*). A better algorithm for assigning semantic classes and a more refined semantic class hierarchy are needed.

(18) **(Metaphor)**$_1$, a software subsidiary that IBM purchased in 1991, also named Chris Grejtak, ... Mr. Grejtak said in an interview that the staff reductions will affect most areas of **(the company)**$_2$ related to its early proprietary software products.

(19) Business brief—Petrie Stores Corp.: Losses for **(Fiscal 2nd Period)**$_1$, half seen likely by retailer.... Petrie Stores Corp., Secaucus, N.J., said an uncertain economy and faltering sales probably will result in a **(second-quarter)**$_2$ loss and perhaps a deficit for the first six months of fiscal 1994.

5.2.4 Errors in Part-of-Speech Assignment. This class of errors is caused by the wrong assignment of part-of-speech tags to words. In (20), (*there*)$_2$ is not extracted because the part-of-speech tag assigned is "RB," which is an adverb and not a possible noun phrase.

(20) Jon W. Slangerup, who is 43 and has been director of customer service in **(Canada)**$_1$, succeeds Mr. Bronczek as vice president and general manager **(there)**$_2$.

5.2.5 Errors in Apposition Determination. This class of errors was described in Section 5.1.4.

5.2.6 Errors in Tokenization. This class of errors is due to the incorrect tokenization of words. In (21), (*1-to-2*)$_1$ and (*15-to-1*)$_2$ are not found because the tokenizer breaks *1-to-2* into *1*, *-*, *to-2*. *15-to-1* is broken up similarly.

(21) Separately, MGM said it completed a previously announced financial restructuring designed to clean up its balance sheet—removing $900 million in bank debt from MGM's books and reducing its debt-to-equity ratio to **(1-to-2)**$_1$ from **(15-to-1)**$_2$—with a view toward a future sale of the company.

5.3 Comparing Errors Made by RESOLVE

McCarthy (1996) has also performed an analysis of errors while conducting an evaluation on the MUC-5 English Joint Venture (EJV) corpus. A large number of the spurious links are caused by what he terms "feature ambiguity," which means that feature values are not computed perfectly. As seen in Table 5, our string match feature accounts for most of the spurious links. Also, seven of the spurious links are caused by alias and apposition determination. As with RESOLVE, "feature ambiguity" is the main source of precision errors.

For RESOLVE, a large number of the missing links are caused by "incomplete semantic knowledge" (32%) and "unused features" (40.5%). For our system, the errors due to the inadequacy of surface features and semantic class determination problems account for about 75% of the missing links. "Unused features" means that some of the features, or combinations of features, that are needed to classify pairs of phrases as coreferential are not present in the decision trees (McCarthy 1996). Similarly, the inadequacy of our system's surface features means that the current feature set may not be enough and more information sources should be added.

Because a detailed error analysis of RESOLVE would require not only its MUC-6 response file, but also the output of its various components, we cannot perform the same error analysis that we did for our system on RESOLVE.

6. Related Work

There is a long tradition of work on coreference resolution within computational linguistics, but most of it was not subject to empirical evaluation until recently. Among the papers that have reported quantitative evaluation results, most are not based on learning from an annotated corpus (Baldwin 1997; Kameyama 1997; Lappin and Leass 1994; Mitkov 1997).

To our knowledge, the research efforts of Aone and Bennett (1995), Ge, Hale, and Charniak (1998), Kehler (1997), McCarthy and Lehnert (1995), Fisher et al. (1995), and McCarthy (1996) are the only ones that are based on learning from an annotated corpus. Ge, Hale, and Charniak (1998) used a statistical model for resolving pronouns, whereas we used a decision tree learning algorithm and resolved general noun phrases, not just pronouns. Similarly, Kehler (1997) used maximum entropy modeling to assign a probability distribution to alternative sets of coreference relationships among noun phrase entity templates, whereas we used decision tree learning.

The work of Aone and Bennett (1995), McCarthy and Lehnert (1995), Fisher et al. (1995), and McCarthy (1996) employed decision tree learning. The RESOLVE system is presented in McCarthy and Lehnert (1995), Fisher et al. (1995), and McCarthy (1996). McCarthy and Lehnert (1995) describe how RESOLVE was tested on the MUC-5 English Joint Ventures (EJV) corpus. It used a total of 8 features, 3 of which were specific to the EJV domain. For example, the feature JV-CHILD-*i* determined whether *i* referred to a joint venture formed as the result of a tie-up. McCarthy (1996) describes how the original RESOLVE for MUC-5 EJV was improved to include more features, 8 of which were domain specific, and 30 of which were domain independent. Fisher et al. (1995) adapted RESOLVE to work in MUC-6. The features used were slightly changed for this domain. Of the original 30 domain-independent features, 27 were used. The 8 domain-specific features were completely changed for the MUC-6 task. For example, JV-CHILD-*i* was changed to CHILD-*i* to decide whether *i* is a "unit" or a "subsidiary" of a certain parent company. In contrast to RESOLVE, our system makes use of a smaller set of 12 features and, as in Aone and Bennett's (1995) system, the features used are generic and applicable across domains. This makes our coreference engine a domain-independent module.

Although Aone and Bennett's (1995) system also made use of decision tree learning for coreference resolution, they dealt with Japanese texts, and their evaluation focused only on noun phrases denoting organizations, whereas our evaluation, which dealt with English texts, encompassed noun phrases of all types, not just those denoting organizations. In addition, Aone and Bennett evaluated their system on noun phrases that had been correctly identified, whereas we evaluated our coreference resolution engine as part of a total system that first has to identify all the candidate noun phrases and has to deal with the inevitable noisy data when mistakes occur in noun phrase identification and semantic class determination.

The contribution of our work lies in showing that a learning approach, when evaluated on common coreference data sets, is able to achieve accuracy competitive with that of state-of-the-art systems using nonlearning approaches. It is also the first machine learning-based system to offer performance comparable to that of nonlearning approaches.

Finally, the work of Cardie and Wagstaff (1999) also falls under the machine learning approach. However, they used unsupervised learning and their method did not require any annotated training data. Their clustering method achieved a balanced F-measure of only 53.6% on MUC-6 test data. This is to be expected: supervised learning in general outperforms unsupervised learning since a supervised learning algorithm

has access to a richer set of annotated data to learn from. Since our supervised learning approach requires only a modest number of annotated training documents to achieve good performance (as can be seen from the learning curves), we argue that the better accuracy achieved more than justifies the annotation effort incurred.

7. Conclusion

In this paper, we presented a learning approach to coreference resolution of noun phrases in unrestricted text. The approach learns from a small, annotated corpus and the task includes resolving not just pronouns but general noun phrases. We evaluated our approach on common data sets, namely, the MUC-6 and MUC-7 coreference corpora. We obtained encouraging results, indicating that on the general noun phrase coreference task, the learning approach achieves accuracy comparable to that of non-learning approaches.

Acknowledgments

This paper is an expanded version of a preliminary paper that appeared in the *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. We would like to thank the MUC organizers who made available to us the MUC-6 and MUC-7 data sets, without which this work would have been impossible. We also thank Beth Sundheim for helpful comments on an earlier version of this paper, and Hai Leong Chieu for his implementation of the HMM-based named entity recognition module.

References

Aone, Chinatsu and Scott 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*, 122–129.

Baldwin, Breck. 1997. CogNIAC: High precision coreference with limited knowledge and linguistic resources. In *Proceedings of the ACL Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 38–45.

Bikel, Daniel M., Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34(1–3):211–231, February.

Cardie, Claire and Kiri Wagstaff. 1999. Noun phrase coreference as clustering. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 82–89.

Chinchor, Nancy A. 1998. Overview of MUC-7/MET-2. In *Proceedings of the Seventh Message Understanding Conference (MUC-7).* http://www.itl.nist.gov/iad/894.02/related_projects/muc/proceedings/muc_7_toc.html.

Church, Kenneth. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143.

Dietterich, Thomas G. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924, October.

Fisher, David, Stephen Soderland, Joseph McCarthy, Fangfang Feng, and Wendy Lehnert. 1995. Description of the UMass system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 127–140.

Ge, Niyu, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–170.

Kameyama, Megumi. 1997. Recognizing referential links: An information extraction perspective. In *Proceedings of the ACL Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 46–53.

Kehler, Andrew. 1997. Probabilistic coreference in information extraction. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 163–173.

Lappin, Shalom and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, December.

McCarthy, Joseph F. 1996. *A Trainable Approach to Coreference Resolution for Information Extraction*. Ph.D. thesis,

University of Massachusetts Amherst, Department of Computer Science, September.
McCarthy, Joseph F. and Wendy Lehnert. 1995. Using decision trees for coreference resolution. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1050–1055.
McEnery, A., I. Tanaka, and S. Botley. 1997. Corpus annotation and reference resolution. In *Proceedings of the ACL Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 67–74.
Miller, George A. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.
Mitkov, Ruslan. 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 Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 14–21.
MUC-6. 1995. Coreference task definition (v2.3, 8 Sep 95). In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 335–344.
MUC-7. 1997. Coreference task definition (v3.0, 13 Jul 97). In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.
Quinlan, John Ross. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, CA.
Sundheim, Beth M. 1995. Overview of results of the MUC-6 evaluation. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 13–31.