Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, January 2007, pp. 1689--1694.

Shallow Semantics for Coreference Resolution

Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083
vince@hlt.utdallas.edu

Abstract

This paper focuses on the linguistic aspect of noun phrase coreference, investigating the knowledge sources that can potentially improve a learning-based coreference resolution system. Unlike traditional, knowledge-lean coreference resolvers, which rely almost exclusively on morpho-syntactic cues, we show how to induce features that encode semantic knowledge from labeled and unlabeled corpora. Experiments on the ACE data sets indicate that the addition of these new semantic features to a coreference system employing a fairly standard feature set significantly improves its performance.

1 Introduction

Recent years have seen an intensifying interest in noun phrase (NP) coreference — the problem of determining which NPs refer to the same real-world entity in a document, owing in part to the Automatic Content Extraction (ACE) evaluations initiated by NIST as well as the surge of interest in *structured prediction* in the machine learning community. As a result, various new models and approaches to NP coreference are developed. For instance, coreference has been recast as the problem of finding the best path from the root to a leaf in a Bell tree [Luo *et al.*, 2004], and tackled both as a *relational learning* task (see McCallum and Wellner [2004]) and as a *supervised clustering* task (see Li and Roth [2005]).

Equally important to the development of new coreference models is the investigation of new linguistic features for the problem. However, until recently, research in anaphora and coreference resolution has largely adopted a *knowledge-lean* approach, in which resolvers operate by relying on a set of morpho-syntactic cues. While these knowledge-lean approaches have been reasonably successful (see Mitkov *et al.* [2001]), Kehler *et al.* [2004] speculate that deeper linguistic knowledge needs to be made available to resolution systems in order to reach the next level of performance. In fact, it should not be surprising that certain coreference relations cannot be identified by using string-matching facilities and syntactic knowledge alone. For instance, semantic knowledge is needed to determine the coreference relation between two lexically dissimilar common nouns (e.g., *talks* and *negotiations*), and world knowledge might be required to resolve *the president* to *George W. Bush* if such background information was not provided explicitly in the associated document.

Our goal in this paper is to improve the performance of a learning-based coreference system by introducing features that encode semantic knowledge as well as knowledge that is potentially useful for identifying non-anaphoric NPs (i.e., NPs that do not have an antecedent and hence do not need to be resolved). To evaluate the utility of the new linguistic features, we augment a baseline feature set (which comprises knowledge sources commonly employed by existing coreference engines) with these new features. In an evaluation on the ACE datasets, the coreference system using the augmented feature set yields a statistically significant improvement of 2.2-2.3% in F-measure over the baseline system.

Another contribution of our work lies in the use of corpus-based methods for inducing features for coreference resolution. Although there have been a few attempts on inducing gender [Ge *et al.*, 1998], path coreference [Bergsma and Lin, 2006], NP anaphoricity [Bean and Riloff, 1999], and selectional preferences [Dagan and Itai, 1990; Yang *et al.*, 2005] for coreference resolution, most of the existing coreference resolvers rely on heuristic methods for feature computation.

2 New Features for Coreference Resolution

In this section we describe the new linguistic features for our learning-based coreference resolution system.

2.1 Inducing a Semantic Agreement Feature

A feature commonly employed by coreference resolvers for determining whether two NPs are coreferent is the *semantic class* (SC) *agreement* feature, which has the value *true* if the SCs of two NPs match and *false* otherwise. The accuracy of the SC agreement feature, therefore, depends on whether the SC values of the two NPs are computed correctly. For a named entity (NE), the SC is typically determined using an NE recognizer; on the other hand, determining the SC of a common noun proves more difficult, in part because many words are polysemous and it is non-trivial to determine which sense corresponds to the intended meaning of the noun.

To determine the SC of a common noun, many existing coreference systems use WordNet (e.g., Soon *et al.* [2001]), simply assigning to the noun the first (i.e., most frequent) WordNet sense as its SC. It is not easy to measure the accuracy of this heuristic, but the fact that the SC agreement

PERSON, ORGANIZATION, TIME, DAY, MONEY, PERCENT, MEASURE, ABSTRACTION, PSYCHOLOGICAL FEATURE, PHENOMENON, STATE, GROUP, OBJECT, UNKNOWN	

Table 1: List of the possible semantic class values of a common noun returned by the first-sense heuristic method.

feature was not used by Soon *et al.*'s decision tree coreference classifier seems to suggest that the SC values of the NPs were not computed accurately by this "first-sense" heuristic.

Motivated by related work on semantic lexicon construction (e.g., Hearst [1992], Phillips and Riloff [2002]), we develop the following method for learning the SC of a common noun, with the goal of improving the accuracy of the SC agreement feature. Given a large, unannotated corpus[1], we use (1) an in-house NE recognizer (which achieves an F-measure of 93% on the MUC-6 test set) to label each NE with its semantic class, and (2) Lin's [1998b] MINIPAR dependency parser to extract all the appositive relations. An example extraction would be <*Eastern Airlines*, *the carrier*>, where the first entry is a proper noun labeled with either one of the seven MUC-style NE types[2] or OTHERS[3] and the second entry is a common noun. If the proper noun is not labeled as OTHERS, we may infer the SC of the common noun from that of the proper noun. However, since neither MINIPAR nor the NE recognizer is perfect, we use a more robust method for inferring the SC of a common noun: (1) we compute the probability that the common noun co-occurs with each of the eight NE types[4] based on the extracted appositive relations, and (2) if the most likely NE type has a co-occurrence probability above a certain threshold (we set the threshold to 0.7), we label the common noun with the most likely NE type.

An examination of the induced SC values indicates that our method fixes some of the errors commonly made by the first-sense heuristic. For instance, common nouns such as *carrier* and *manufacturer* are typically used to refer to organizations in news articles, but were labeled as PERSON by the heuristic.

Nevertheless, our method has a potential weakness: common nouns that do not belong to any of the seven NE semantic classes remain unlabeled. To address this problem, we will set the SC of an unlabeled common noun to be the value returned by the first-sense heuristic. (In our implementation of the first-sense heuristic, we determine which of the 14 SCs listed in Table 1 a common noun belongs to based on the first WordNet sense.) However, we expect that our method will be able to label most of the common nouns, because in ACE we are primarily interested in nouns referring to a person, organization, or location, as we will see in the next subsection.

2.2 Inducing an ACE-Specific Semantic Feature

The SEM_CLASS feature described in the previous subsection was developed for use in a general-purpose coreference system. However, because of the way the ACE coreference task is defined, we may be able to improve system performance on the ACE data if we develop another semantic class agreement feature with the ACE guidelines in mind. Specifically, the ACE coreference task is concerned with resolving references to NPs belonging to one of the five *ACE semantic classes* (ASCs), namely, PERSON, ORGANIZATION, FACILITY, GSP (a geographical-social-political region), and LOCATION [see Table 2 for a brief description of the ASCs]. In particular, references to NPs belonging to other SCs are not to be marked up. Hence, we desire an ACE_SEM_CLASS feature that considers two NPs semantically compatible if and only if the two NPs have a common ASC. The rest of this subsection describes how we determine the ASC of an NP. As we will see, we allow an NP to possess more than one ASC in some cases.

PERSON	human
ORGANIZATION	corporation, agency, government
FACILITY	man-made structure (e.g., building)
GSP	geo-political region (e.g., country, city)
LOCATION	geographical area and landmass, body of water, geological formation

Table 2: ACE semantic classes.

ORGANIZATION	social group
FACILITY	establishment, construction, building, facility, workplace
GSP	country, province, government, town, city, administration, society, island, community
LOCATION	dry land, region, landmass, body of water, geographical area, geological formation

Table 3: List of keywords used in WordNet search for determining the ACE semantic class of a common noun.

Our method for determining the ASC of an NP is based in part on its SC value as computed by the SEM_CLASS feature. In particular, the method hinges on the observation that (1) SEM_CLASS's ORGANIZATION class roughly corresponds to two ASCs: FACILITY and ORGANIZATION, and (2) SEM_CLASS's LOCATION class roughly corresponds to two ASCs: GSP and LOCATION. Given this observation, we can determine the ASC of an NP as follows:

- If its SEM_CLASS value is not PERSON, ORGANIZATION, or LOCATION, its ASC will be OTHERS.
- If its SEM_CLASS is PERSON, its ASC will be PERSON.
- If its SEM_CLASS is LOCATION, we will have to determine whether its ASC is LOCATION or GSP, according to our observation above. Specifically, we first use WordNet to determine whether the head noun of the NP is a hypernym of one of the GSP keywords listed in Table 3.[5] We then repeat this WordNet lookup procedure using the LOCATION keywords. If both lookups are successful, the ASC of the NP will be both GSP and LOCATION; otherwise the ASC will be one of these two classes.
- If its SEM_CLASS is ORGANIZATION, we will have to determine whether its ASC is ORGANIZATION or FA-

[1] We used (1) the BLLIP corpus (30M words), which consists of Wall Street Journal articles from 1987 to 1989, and (2) the Reuters Corpus (3.7GB data), which has 806,791 Reuters articles.

[2] Person, organization, location, date, time, money, and percent.

[3] This indicates the proper noun is not a MUC NE.

[4] For simplicity, OTHERS is viewed as an NE type here.

[5] The keywords are obtained via our experimentation with WordNet and the ASCs of the NPs in the training data.

CILITY. We can similarly use the procedure outlined in the previous bullet to determine whether its ASC is ORGANIZATION, FACILITY, or both.

2.3 Inducing a Semantic Similarity Feature

Many reference resolvers use WordNet to compute the semantic similarity between two common nouns (e.g., Poesio *et al.* [2004] and Daumé and Marcu [2005]). However, this approach to determining semantic similarity may not be robust, since its success depends to a large extent on the ability to determine the correct WordNet sense of the given nouns.

Motivated by research in lexical semantics, we instead adopt a *distributional* approach to computing the semantic similarity between two common nouns: we capture the semantics of a noun by counting how frequent it co-occurs with other words, determining a pair of common nouns to be semantically similar if their co-occurrence patterns are similar.

Instead of acquiring semantic similarity information from scratch, we use the semantic similarity values provided by Lin's [1998a] dependency-based thesaurus, which is constructed using a distributional approach combined with an information-theoretic definition of similarity. Each word w in the thesaurus is associated with a list of words most similar to w together with the semantic similarity values.

Given the thesaurus, we can construct a semantic similarity feature, SEM_SIM, for coreference resolution, in which we use a binary value to denote whether two NPs, NP_x and NP_y, are semantically similar. Specifically, the feature has the value *true* if and only if NP_x is among the 5-nearest neighbors of NP_y according to the thesaurus or vice versa.

2.4 Inducing a Pattern-Based Feature

Next, we induce a PATTERN_BASED feature using information provided by an algorithm that learns patterns for extracting coreferent NP pairs, each of which involves a pronoun and its antecedent. Bean and Riloff [2004] also learn extraction patterns for coreference resolution, but unlike our method, their method is unsupervised and domain-specific.

Before showing how to compute our PATTERN_BASED feature, let us describe the pattern learner, which operates as follows: (1) patterns are acquired from a corpus annotated with coreference information, and (2) the accuracy of each learned pattern is estimated. Below we elaborate these two steps.

Acquiring the patterns. Recall that a pattern is used to extract coreferent NP pairs; hence a good pattern should capture features of the two NPs involved as well as the context in which they occur. To illustrate how to induce a pattern, let us consider the following *coreference segment* (CS), which we define as a text segment that starts with an NP, NP_x, and ends with a pronoun that co-refers with NP_x: "John is studying hard for the exam. He". From this CS, we can induce a pattern that simply comprises all the tokens in the segment. If we see this sequence of tokens in a test text, we can apply this pattern to determine that *John* and *He* are likely to co-refer.

The above pattern, however, may not be useful because it is unlikely that we will see exactly the same text segment in an unseen text. Hence, we desire a pattern learner that can generalize from a CS and yet retain sufficient information to extract coreferent NP pairs. Specifically, we design a pattern learner that induces from each CS in a given annotated corpus three extraction patterns, each of which represents a different degree of generalization from the CS. In this work, we only consider segments in which the antecedent and the anaphor are fewer than three sentences apart.

Table 4 shows the three patterns that the learner will induce for the example CS above. The first pattern (see row 1) is created by (1) representing the antecedent and the anaphor as a set of attribute values indicating its gender, number, semantic class, grammatical role, and NP type[6]; (2) representing each of the remaining NPs in the CS by the token NP; (3) representing each non-verbal and non-adverbial token that is not enclosed by any NP by its part-of-speech tag; and (4) representing each verbal token as it is. The reason for retaining the verbal tokens is motivated by the intuition that verbs can sometimes play an important role in identifying coreferent NP pairs. On the other hand, adverbs are not represented because they generally do not contain useful information as far as identifying coreferent NP pairs is concerned.

The second pattern (shown in row 2 of Table 4) is created via the same procedure as the first pattern, except that each verbal token is replaced by its part-of-speech tag. The third pattern (see row 3 of Table 4) is created via the same procedure as the preceding two patterns, except that only the NPs are retained. So, the three patterns represent three different levels of generalizations from the CS, with the first one being the most specific and the third one being the most general.

Note that some of the induced patterns may extract both coreferent *and* non-coreferent NP pairs, thus having a low extraction accuracy. The reason is that our pattern learner does not capture evidence outside a CS segment, which in some cases may be crucial for inducing high-precision rules. Hence, we need to estimate the accuracy of each pattern, so that a coreference system can decide whether it should discard NP pairs extracted by patterns with a low accuracy.

Estimating pattern accuracy. After we acquire a list of extraction patterns L (with the redundant patterns removed), we can estimate the *accuracy* of each pattern on an annotated corpus simply by counting the number of coreferent NP pairs it extracts divided by the total number of NP pairs it extracts, as described below. First, we collect all the CS's and *non-coreference segments* (NCS's) from the same annotated corpus that we used to induce our patterns, where an NCS is defined as a text segment beginning with an NP, NP_x, and ends with a pronoun, NP_y, such that NP_x is *not* coreferent with NP_y. (As before, we only consider CS's and NCS's where the two enclosing NPs are separated by fewer than three sentences.) From each CS/NCS, we use our pattern learner to induce three patterns in the same way as before, but this time we additionally label each pattern with a '-' if it was induced from an NCS and a '+' otherwise. We then insert all these labeled patterns into a list S, with the redundant patterns retained. Finally, we compute the accuracy of a pattern $l \in L$ as the number of times l appears in S with the label '+' divided by the total number of times l appears in S.

[6]The possible NP types are PRONOUN, PROPER_NOUN, and COMMON_NOUN.

1.	{ GENDER:M, NUMBER:SING, SEMCLASS:PERSON, GRAMROLE:SUBJ, NPTYPE:PN } is studying IN NP . { GENDER:M, NUMBER:SING, SEMCLASS:PERSON, GRAMROLE:SUBJ, NPTYPE:PRO }
2.	{ GENDER:M, NUMBER:SING, SEMCLASS:PERSON, GRAMROLE:SUBJ, NPTYPE:PN } VBZ VBG IN NP . { GENDER:M, NUMBER:SING, SEMCLASS:PERSON, GRAMROLE:SUBJ, NPTYPE:PRO }
3.	{ GENDER:M, NUMBER:SING, SEMCLASS:PERSON, GRAMROLE:SUBJ, NPTYPE:PN } NP { GENDER:M, NUMBER:SING, SEMCLASS:PERSON, GRAMROLE:SUBJ, NPTYPE:PRO }

Table 4: The three patterns induced for the coreference segment "John is studying hard for the exam. He".

Using the list of extraction patterns L sorted in decreasing order of accuracy, we can create our PATTERN_BASED feature as follows. Given a pair of NPs, we march down the pattern list L and check if any of the patterns can extract the two NPs. If so, the feature value is the accuracy of the *first* pattern that extracts the two NPs; otherwise the feature value is 0.

2.5 Inducing an Anaphoricity Feature

Anaphoricity determination refers to the problem of determining whether an NP has an antecedent or not. Knowledge of anaphoricity can potentially be used to identify and filter non-anaphoric NPs prior to coreference resolution, thereby improving the precision of a coreference system. There have been attempts on identifying non-anaphoric phrases such as pleonastic *it* (e.g., Lappin and Leass [1994]) and non-anaphoric definite descriptions (e.g., Bean and Riloff [1999]).

Unlike previous work, our goal here is *not* to build a full-fledged system for identifying and filtering non-anaphoric NPs. Rather, we want to examine whether *shallow* anaphoricity information, when encoded as a feature, could benefit a learning-based coreference system. Specifically, we employ a simple method for inducing anaphoricity information: given a corpus labeled with coreference information, we compute the anaphoricity value of an NP, NP_x, as the probability that NP_x has an antecedent in the corpus. If NP_x never occurs in the annotated corpus, we assign to it the default anaphoricity value of -1. Hence, unlike previous work, we represent anaphoricity as a real value rather than a binary value.

Now we can encode anaphoricity information as a feature for our learning-based coreference system as follows. Given a coreference instance involving NP_x and NP_y, we create a feature whose value is simply NP_y's anaphoricity value.

Conceivably, data sparseness may render our ANAPHORICITY feature less useful than we desire. However, a glimpse at the anaphoricity values computed by this feature shows that it can capture some potentially useful information. For instance, the feature encodes that *it* only has a moderate probability of being anaphoric, and the NP *the contrary* taken from the phrase *on the contrary* is never anaphoric.

2.6 Inducing a Coreferentiality Feature

We can adapt the above method for generating the anaphoricity feature to create a COREFERENTIALITY feature, which encodes the probability that two NPs are coreferent. These coreferentiality probabilities can again be estimated from a corpus annotated with coreference information; in cases where one or both of the given NPs do not appear in the corpus, we set the coreferentiality value of the NP pair to -1.

Our method of inducing the COREFERENTIALITY feature may also suffer from data sparseness. However, whether this feature is useful at all for coreference resolution is an empirical question, and we will evaluate its utility in Section 4.

3 The Baseline Feature Set

The previous section introduced our new features for coreference resolution. As mentioned in the introduction, these features will be used in combination with a set of baseline features. This section describes our baseline feature set, which comprises 34 selected features employed by high-performing coreference systems such as Soon *et al.* [2001], Ng and Cardie [2002], and Ponzetto and Strube [2006].

Lexical features. We use nine features to allow different types of string matching operations to be performed on the given pair of NPs, NP_x and NP_y[7], including (1) exact string match for pronouns, proper nouns, and non-pronominal NPs (both before and after determiners are removed); (2) substring match for proper nouns and non-pronominal NPs; and (3) head noun match. In addition, a nationality matching feature is used to match, for instance, *British* with *Britain*. Furthermore, we have a feature that tests whether all the words that appear in one NP also appear in the other NP.

Grammatical features. 23 features test the grammatical properties of one or both of the NPs. These include ten features that test whether each of the two NPs is a pronoun, a definite NP, an indefinite NP, a nested NP, and a clausal subject. A similar set of five features is used to test whether both NPs are pronouns, definite NPs, nested NPs, proper nouns, and clausal subjects. In addition, five features determine whether the two NPs are compatible with respect to gender, number, animacy, and grammatical role. Furthermore, two features test whether the two NPs are in apposition or participate in a predicate nominal construction (i.e., the IS-A relation). Finally, motivated by Soon *et al.* [2001], we have a feature that determines whether NP_y is a demonstrative NP.

Semantic features. There are two semantic features, both of which are employed by Soon *et al.*'s coreference system. The first feature tests whether the two NPs have the same semantic class. Here, the semantic class of a proper noun and a common noun is computed using an NE finder and WordNet (by choosing the first sense), respectively. The second feature tests whether one NP is a name alias or acronym of the other.

Positional features. We have one positional feature that measures the distance between the two NPs in sentences.

4 Evaluation

In this section, we evaluate the effectiveness of our newly proposed features in improving the baseline coreference system.

[7] We assume that NP_x precedes NP_y in the associated text.

4.1 Experimental Setup

We use the ACE-2 (Version 1.0) coreference corpus for evaluation purposes. The corpus is composed of three data sets taken from three different news sources: Broadcast News (BR), Newspaper (PA), and Newswire (WI). Each data set comprises a set of training texts for acquiring coreference classifiers and a set of test sets for evaluating the output of the coreference system. We report performance in terms of recall, precision, and F-measure using two different scoring programs: the commonly-used MUC scorer [Vilain *et al.*, 1995] and the recently-developed CEAF scorer [Luo, 2005]. According to Luo, CEAF is designed to address a potential problem with the MUC scorer: partitions in which NPs are over-clustered tend to be under-penalized. For all of the experiments conducted in this paper, we use NPs automatically extracted by an in-house NP chunker and an NE recognizer.

4.2 The Baseline Coreference System

Our baseline system uses the C4.5 decision tree learning algorithm [Quinlan, 1993] in conjunction with the 34 baseline features described in Section 3 to acquire a coreference classifier on the training texts for determining whether two NPs are coreferent. To create training instances, we pair each NP in a training text with each of its preceding NPs, labeling an instance as positive if the two NPs are in the same coreference chain in the associated text and negative otherwise.

After training, the decision tree classifier is used to select an antecedent for each NP in a test text. Following Soon *et al.* [2001], we select as the antecedent of each NP, NP_j, the *closest* preceding NP that is classified as coreferent with NP_j. If no such NP exists, no antecedent is selected for NP_j.

Row 1 of Table 5 and Table 6 shows the results of the baseline system obtained via the MUC scorer and the CEAF scorer, respectively. Each row of the two tables corresponds to an experiment evaluated on four different test sets: the entire ACE test set (comprising all the BR, PA, and WI test texts) and each of the BR, PA, and WI test sets. These four sets of results are obtained by applying the same coreference classifier that is trained on the entire ACE training corpus (comprising all the training texts from PA, WI, and BR). Owing to space limitations, we will mainly discuss results obtained on the entire test set. As we can see, the baseline achieves an F-measure of 62.0 (MUC) and 60.0 (CEAF).

To get a better sense of how strong these baseline results are, we repeat the above experiment except that we replace the 34 features with the 12 features employed by Soon *et al.*'s [2001] coreference resolver. Results of the Soon *et al.* system, shown in row 2 of the two tables, indicates that our baseline features yield significantly better results than Soon *et al.*'s[8]: F-measure increases by 5.4 (MUC) and 3.7 (CEAF).

4.3 Coreference Using the Expanded Feature Set

Next, we train a coreference resolver using the baseline feature set augmented with the five new features described in Sections 2.2–2.6, namely, ACE_SEMCLASS, SEM_SIM, PATTERN_BASED, ANAPHORICITY, and COREFERENTIALITY. In addition, we replace the heuristic-based SC agreement feature in the baseline feature set with our SEM_CLASS feature (see Section 2.1). We employ the same methods for training instance creation and antecedent selection as in the baseline.

Recall that the PATTERN_BASED, ANAPHORICITY, and COREFERENTIALITY features are all computed using a data set annotated with coreference information. Hence, we need to reserve a portion of our training texts for the purpose of computing these features. Specifically, we partition the available training texts into two sets of roughly the same size: the training subset and the development subset. The development subset will be used for computing those features that require an annotated corpus, and the training subset will be used to train the coreference classifier using the expanded feature set.

Results using the expanded feature set are shown in row 3 of the two tables. In comparison to the baseline results in row 1, we see that F-measure increases from 62.0 to 64.2 (MUC) and 60.0 to 62.3 (CEAF). Although the gains may seem moderate, the performance difference as measured by both scorers is in fact highly statistically significant, with p=0.0004 for MUC and p=0.0016 for CEAF.

4.4 Feature Analysis

To better understand which features are important for coreference resolution, we examine the decision tree learned using the expanded feature set (not shown here due to space limitations). At the top of the tree are the two lexical features that test exact string match for proper nouns and for non-pronominal NPs. This should not be surprising, since these string matching features are generally strong indicators of coreference. Looking further down the tree, we see the SEM_CLASS, ANAPHORICITY, and COREFERENTIALITY features appearing in the third and fourth levels of the tree. This indicates that these three features play a significant role in determining whether two NPs are coreferent.

To further investigate the contribution of each of our new features to overall performance, we remove each new feature (one at a time) from the expanded feature set and re-train the coreference classifier using the remaining features. Results are shown in rows 4–9 of Tables 5 and 6, where an asterisk (*) is used to indicate that the corresponding F-measure is significantly different from that in row 3 (at p=0.05). From these results, we make two observations. First, removing ANAPHORICITY, COREFERENTIALITY or ACE_SEM_CLASS precipitates a significant drop in F-measure, whichever scoring program is used. Interestingly, even though we are faced with data sparseness when computing ANAPHORICITY and COREFERENTIALITY, both features turn out to be useful. Second, although removing SEM_CLASS does not result in a significant drop in performance, it does not imply that SEM_CLASS is not useful. In fact, as mentioned at the beginning of this subsection, SEM_CLASS appears near the top of the tree. An inspection of the relevant decision tree reveals that the learner substitutes ACE_SEM_CLASS for SEM_CLASS when the latter feature is absent. This explains in part why we do not see a large drop in F-measure. Hence, we can only claim that SEM_CLASS is not important in the presence of ACE_SEM_CLASS, a feature with which it is correlated.

[8] Like the MUC organizers, we use Noreen's [1989] Approximate Randomization method for significance testing, with p set to 0.05.

	Experiments	Entire Test Set R	P	F	Broadcast News (BR) R	P	F	Newspaper (PA) R	P	F	Newswire (WI) R	P	F
1	Using the baseline features only	53.7	73.4	62.0	53.6	72.7	61.7	55.3	74.7	63.5	52.2	72.9	60.8
2	Using Soon *et al.*'s features only	46.2	73.2	56.6	43.8	70.4	54.0	50.0	75.9	60.3	44.6	73.0	55.4
3	Using the expanded feature set	54.7	77.8	64.2	56.1	76.3	64.7	54.4	79.7	64.6	53.5	77.5	**63.3**
4	without SEM_CLASS	55.1	77.5	64.4	56.1	75.7	64.5	55.3	79.5	65.3	53.6	77.3	**63.3**
5	without ACE_SEM_CLASS	53.4	77.1	63.1*	55.2	76.2	64.0	54.0	79.2	64.2	50.9	75.9	60.9*
6	without SEM_SIM	54.7	77.6	64.2	56.1	76.1	64.6	54.7	79.6	64.8	53.2	77.3	63.0
7	without PATTERN_BASED	55.0	77.8	**64.5**	57.2	76.3	**65.4***	54.6	80.0	**65.0**	53.2	76.7	62.8
8	without ANAPHORICITY	53.7	77.8	63.5*	54.0	76.6	63.4*	54.8	79.5	64.9	52.2	77.3	62.3
9	without COREFERENTIALITY	53.7	78.3	63.3*	54.3	76.5	63.5	53.7	80.9	64.6	51.2	77.7	61.7

Table 5: Results obtained via the MUC scorer by learning coreference classifiers from the entire ACE training corpus.

	Experiments	Entire Test Set R	P	F	Broadcast News (BR) R	P	F	Newspaper (PA) R	P	F	Newswire (WI) R	P	F
1	Using the baseline features only	55.4	65.4	60.0	56.8	66.0	61.1	53.9	63.5	58.3	55.5	66.8	60.7
2	Using Soon *et al.*'s features only	49.8	64.9	56.3	49.3	63.7	55.6	50.0	64.4	56.3	50.2	66.7	57.3
3	Using the expanded feature set	56.7	69.0	**62.3**	57.3	66.9	61.7	55.1	69.5	61.5	57.7	70.9	**63.6**
4	without SEM_CLASS	56.1	67.9	61.4	57.1	66.3	61.4	53.7	67.2	59.8	57.2	70.4	63.1
5	without ACE_SEM_CLASS	54.6	67.2	60.2*	56.3	66.6	61.0	52.5	66.3	58.6*	55.9	68.9	61.1*
6	without SEM_SIM	56.4	68.1	61.7	57.1	66.5	61.5	55.4	69.7	**61.8**	57.2	70.1	63.0
7	without PATTERN_BASED	56.2	68.2	61.6	58.0	67.5	**62.4**	53.4	67.6	59.7	57.0	69.7	62.7
8	without ANAPHORICITY	55.0	67.9	60.8*	55.6	66.6	60.6	54.8	69.1	61.1	54.6	68.0	60.6*
9	without COREFERENTIALITY	55.0	68.5	61.0*	55.3	66.3	60.3	53.5	68.1	59.9	56.3	71.3	62.9

Table 6: Results obtained via the CEAF scorer by learning coreference classifiers from the entire ACE training corpus.

5 Conclusions

In this paper, we investigated the relative contribution of our proposed features for learning-based coreference resolution. While we obtained encouraging results on the ACE data sets, we should note that performance gains are limited in part by the difficulty in accurately computing these features given current language technologies. We expect that these features can provide further improvements if we increase the training data and develop even better methods for computing them. As noted before, there have been very few attempts on using corpus-based methods for inducing features for coreference resolution, and we believe our work contributes to the corpus-based induction of semantic and other non-morpho-syntactic features for coreference resolution.

References

[Bean and Riloff, 1999] D. Bean and E. Riloff. Corpus-based identification of non-anaphoric noun phrases. In *Proc. of the ACL*, pages 373–380.

[Bean and Riloff, 2004] D. Bean and E. Riloff. Unsupervised learning of contextual role knowledge for coreference resolution. In *NAACL*.

[Bergsma and Lin, 2006] S. Bergsma and D. Lin. Bootstrapping path-based pronoun resolution. In *Proc. of COLING/ACL*, pages 33–40.

[Dagan and Itai, 1990] I. Dagan and A. Itai. Automatic processing of large corpora for the resolution of anaphora references. In *Proc. of COLING*.

[Daumé and Marcu, 2005] H. Daumé III and D. Marcu. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proc. of HLT/EMNLP*, pages 97–104.

[Ge *et al.*, 1998] N. Ge, J. Hale, and E. Charniak. A statistical approach to anaphora resolution. In *Proc. of WVLC*, pages 161–170.

[Hearst, 1992] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING*, pages 539–545.

[Kehler *et al.*, 2004] A. Kehler, D. Appelt, L. Taylor, and A. Simma. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proc. of HLT/NAACL*, pages 289–296.

[Lappin and Leass, 1994] S. Lappin and H. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4).

[Li and Roth, 2005] X. Li and D. Roth. Discriminative training of clustering functions: Theory and experiments with entity identification. In *CoNLL*.

[Lin, 1998a] D. Lin. Automatic retrieval and clustering of similar words. In *Proc. of COLING/ACL*, pages 768–774.

[Lin, 1998b] D. Lin. Dependency-based evaluation of MINIPAR. In *Proc. of the LREC Workshop on the Evaluation of Parsing Systems*.

[Luo, 2005] X. Luo. On coreference resolution performance metrics. In *Proc. of HLT/EMNLP*, pages 25–32.

[Luo *et al.*, 2004] X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. A mention-synchronous coreference resolution algorithm based on the Bell tree. In *Proc. of the ACL*, pages 153–142.

[McCallum and Wellner, 2004] A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Advances in NIPS*.

[Mitkov *et al.*, 2001] R. Mitkov, B. Boguraev, and S. Lappin. Introduction to the special issue on computational anaphora resolution. *Computational Linguistics*, 27(4):473–477.

[Ng and Cardie, 2002] V. Ng and C. Cardie. Improving machine learning approaches to coreference resolution. In *Proc. of the ACL*.

[Noreen, 1989] E. W. Noreen. 1989. *Computer Intensive Methods for Testing Hypothesis: An Introduction*. John Wiley & Sons.

[Phillips and Riloff, 2002] W. Phillips and E. Riloff. Exploiting strong syntactic heuristics and co-training to learn semantic lexicons. In *EMNLP*.

[Poesio *et al.*, 2004] M. Poesio, R. Mehta, A. Maroudas, and J. Hitzeman. Learning to resolve bridging references. In *Proc. of the ACL*.

[Ponzetto and Strube, 2006] S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proc. of HLT/NAACL*, pages 192–199.

[Quinlan, 1993] J. R. Quinlan. *C4.5: Programs for Machine Learning*.

[Soon *et al.*, 2001] W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

[Vilain *et al.*, 1995] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. A model-theoretic coreference scoring scheme. In *MUC-6*.

[Yang *et al.*, 2005] X. Yang, J. Su, and C. L. Tan. Improving pronoun resolution using statistics-based semantic compatibility information. In *ACL*.