Accurate Semantic Class Classifier for Coreference Resolution

Zhiheng Huang[1], **Guangping Zeng**[1,2], **Weiqun Xu**[3], **and Asli Celikyilmaz**[1]
[1]EECS Department, University of California at Berkeley,
CA 94720, USA
{zhiheng,gpzeng,asli}@eecs.berkeley.edu
[2]Computer Science Department, School of Information Engineering,
University of Science and Technology Beijing, China
[3]ThinkIT, Institute of Acoustics, Chinese Academy of Sciences,
Beijing, 100190, China
xuweiqun@hccl.ioa.ac.cn

Abstract

There have been considerable attempts to incorporate semantic knowledge into coreference resolution systems: different knowledge sources such as WordNet and Wikipedia have been used to boost the performance. In this paper, we propose new ways to extract WordNet feature. This feature, along with other features such as named entity feature, can be used to build an accurate *semantic class* (SC) classifier. In addition, we analyze the SC classification errors and propose to use relaxed SC agreement features. The proposed accurate SC classifier and the relaxation of SC agreement features on ACE2 coreference evaluation can boost our baseline system by 10.4% and 9.7% using MUC score and anaphor accuracy respectively.

1 Introduction

Coreference resolution is used to determine which noun phrases (including pronouns, proper names, and common nouns) refer to the same entities in documents. Much work on coreference resolution is based on (Soon et al., 2001), which built a decision tree classifier to label pairs of mentions as coreferent or not. Recent work aims to improve the performance from two aspects: new models and new features. The former cast the pair wise mention classifications into various forms such as the best path in a Bell tree (Luo et al., 2004), the best graph cut (Nicolae and Nicolae, 2006), integer linear programming (Denis and Baldridge, 2007) and graph partition based conditional model (McCallum and Wellner, 2004). The latter develop and investigate new linguistic features for the problem. For instance, WordNet (Poesio et al., 2004), Wikipedia (Ponzetto and Strube, 2006), semantic neighbor words (Ng, 2007a), and pattern based features (Yang and Su, 2007) have been extensively studied.

Deeper linguistic knowledge is required to enable the coreference resolution to reach a higher level of performance (Kehler et al., 2004). An important type of semantic knowledge that has been employed in coreference resolution system is the semantic class (SC) of an NP, which can be used to filter out the coreference between semantically incompatible NPs. However, the difficulty is to accurately compute the semantic class features. In this paper, we show that the WordNet may not be efficiently employed in the traditional way such as (Soon et al., 2001; Ng, 2007a; Ponzetto and Strube, 2006) to compute the semantic class features. We introduce new ways to use the WordNet and the experiments show its effectiveness in determining the semantic classes for noun phrases. In addition, we analyze the classification errors of the SC classifier and propose to use relaxed SC agreement features. With these proposed features and other standard syntactic features (which are commonly employed in existing coreference systems), our coreference resolution system can obtain an increase of 10.4% for MUC score and 9.7% for anaphor accuracy from the baseline in ACE2 evaluation.

2 Related Work

WordNet (Fellbaum, 1998) as an important knowledge source has been widely employed in previous coreference resolution work. For example, Harabagiu et al. (2001) have used WordNet relations such as synonym and is-a to mine the patterns of WordNet paths for pairs of antecedents and anaphors. Due to the nature of the rule based coreference system (in contrast to machine learning based), the weights of relations may not be accurately estimated. Vieira and Poesio (2000) and Markert and Nissim (2005) have used WordNet synonym and hyponym etc. to determine if an anaphor semantically relates to one previous NP. Ponzetto and Strube (2006) have used WordNet semantic similarity and relatedness scores between antecedents and candidate anaphors. Their

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1232–1240,
Singapore, 6-7 August 2009. ©2009 ACL and AFNLP

work is different to this work in the following: 1) Their work involves various relations such as hyponyms and meronyms while ours only makes use of hypernyms; and 2) Their work focuses on investigating if two NPs have particular WordNet relations or not, while ours focuses on using WordNet hypernyms for their SC classification and then testing their SC compatibility. In doing so, we can directly model the accuracy of semantic class classification and test its impact on coreference resolution.

While the SC of a proper name is computed fairly accurately using a named entity (NE) recognizer, many coreference resolution systems simply assign to a common noun the first (i.e., most frequent) WordNet synset as its SC (Soon et al., 2001; Markert and Nissim, 2005). This heuristics, apparently, did not lead to good performance. The best reported ACE2 coreference resolution system (Ng, 2007a; Ng, 2007b) has proposed an accurate SC classifier which used heterogeneous semantic knowledge sources. WordNet is just one of the several knowledge sources which have been utilized. However, the WordNet based features is not informative compared to other features such as the semantic neighbor feature. Similarly, Ponzetto and Strube (2006) have discovered that the WordNet feature is no more informative than the community-generated Wikipedia feature. In this paper, we focus on the investigation of various usages of WordNet for the SC classification task. The work which is directly comparable to ours would be (Ng, 2007a; Ng, 2007b).

Other similar work includes the *mention detection* (MD) task (Florian et al., 2006) and joint probabilistic model of coreference (Daumé III and Marcu, 2005). The MD task identifies the boundary of a mention, its mention type (e.g., pronoun, name), and its semantic type (e.g., person, organization). Unlike them, we do not perform the boundary detection, as we make use of the noun phrases directly from the noun phrase chunker and NE recognizer. The joint probabilistic model models the MD and coreference simultaneously, while our work focuses on them separately.

3 Semantic Class Classification

In this section, we describe how we compile the training corpus and extract features using WordNet. We report our results on the ACE coreference corpus due to that it has been commonly used and it was annotated SCs of six types.[1] As in (Ng, 2007a), we first train a classifier to predict the SC of an NP. This SC information is used later in the coreference resolution stage. For example, *the audience* is classified as SC of *person*, and it thus should not be coreferent with *the security industry*, which is usually classified as *organization*. This task is by no means trivial. First, while the classification of *Tom Hanks* being SC of *person* can be accurately achieved by an NE recognizer, the association of *audience* and *person* requires semantic language source such as WordNet. Second, the same noun phrase can be annotated with different SCs under different context. For example, *the authorities* is usually annotated as *person*, but it is sometimes as *organization*. Even worse, the same noun phrases are sometimes annotated with one of the five explicitly annotated classes while sometimes are not annotated at all (thus falling into the other SC). For example, *people* is annotated as *person* SC explicitly 20 times and is not annotated at all 21 times in the ACE2 testset. This inconsistent annotation adversely affects the performance of an SC classifier. And this in turn would cause errors during coreference stage. In section 4.3, we show how to relax the strict SC agreement feature to address this.

3.1 Training instance creation

We use ACE Phase 2 Coreference corpus to train the SC classifier. Each noun phrase which is identified by the noun phrase chunker or NE recognizer is used to create a training instance. Each instance is represented by a set of lexical, syntactic and semantic features, as described below. If the NP under consideration is annotated as one of the five ACE SCs in the corpus, then the classification of the associated training instance is the ACE SC of the NP. Otherwise, the instance is labeled as other.

ACE 2 corpus has a training set and a test set which comprise of 422 and 97 texts respectively. We divide the training set into a new training and a development set: the former consists of 90% randomly generated and stratified original training instances and the latter consists of the rest 10% instances. The test set remains the same as in ACE2 corpus. The size of each dataset and its SC distributions are shown in Table 1. Note that the training and development datasets have exactly the same distributions of SCs due to the stratification procedure. That is, each class has the same proportion in training and development datasets. We tune the feature parameters against development set and report performance on both development set and test set.

[1] Person, organization, gpe, location and facility are explicitly annotated. The rest noun phrases are other type.

Table 1: Distributions of SCs in ACE2 corpus.

	Size	PER	ORG	GPE	FAC	LOC	OTH
Train	55629	20.29	7.30	8.42	0.61	0.55	62.80
Dev	6181	20.29	7.30	8.42	0.61	0.55	62.80
Test	15360	20.48	7.57	6.90	0.85	0.41	63.79

3.2 Lexical features

Each instance is represented as a bag of features and is fed into a classifier in training stage. We present four binary lexical feature sets as follows.

Word unigrams and bigrams: An N-gram is a sub-sequence of N words from a given noun phrase. Unigram forms the bag of words feature, and bigram forms the pairs of words feature, and so forth. We have considered word unigram and bigram features in our experiments.

First and last words: This feature extracts the first and last words of an NP. For example, the first word *the* and the last word *store* are extracted from the NP *the main store*. This feature does not only coarsely models the influence of the first word, for example, *a* or *the*, but also models the head word, since the head word usually is the last word in the NP.

Head word: We use Collins style rules (Collins, 1999) to extract the head words for given NPs. These features should be most informative if the training corpus is large enough.[2] For example, the head word *company* of the NP *the company* immediately determines its SC being *organization*. However, due to the sparseness of training data, its potential importance is adversely affected.

3.3 Semantic features

NE feature is extracted from Stanford named entity recognizer (NER) (Finkel et al., 2005). Three types of named entities: person, location and organization can be recognized for a given NP. This feature is primarily useful for SC classification of proper nouns.

WordNet is a large English lexicon in which semantically related words are connected via cognitive synonyms (synsets). The WordNet is a useful tool for word semantics analysis and has been widely used in natural language processing applications. In WordNet, synsets are organized into hierarchies with hypernym/hyponym relationships: Y is a hypernym of X if every X is a (kind of) Y (X is called a hyponym of Y in this case).

The WordNet is employed in (Ng, 2007a) as following to create the **WN_CLASS** feature. For each keyword w as shown in the right column of Table 2, if the head noun of a given NP is a hyponym of w in WordNet,[3] then the word w becomes a feature for such NP. It is explained that these keywords are correlated with the ACE SCs and they are obtained via experimentation with WordNet and the ACE SCs of the NPs in the ACE training data. However, it is likely that these handcrafted keywords have poor coverage for general cases. As a result, it may not make full use of WordNet semantic knowledge. This will be shown in our individual feature contribution experiment in Section 3.5.

Table 2: List of keywords used in WordNet semantic feature in (Ng, 2007a).

ACE SC	Keywords
PER	person
ORG	social group
FAC	establishment, construction, building, facility, workplace
GPE	country, province, government, town, city, administration, society, island, community
LOC	dry land, region, landmass, body of water geographical area, geological formation

There are other ways of using WordNet for semantic feature extraction. For example, Ponzetto and Strube (2006) have employed WordNet similarity measure for coreference resolution. The difference is that they created the feature directly at the coreference resolution stage, ie, using the WordNet similarity between the antecedent and anaphor to determine if they are coreferent, while we focus on using this feature to classify an NP into a particular SC. For comparison, we implemented a WordNet similarity based feature (**WN_SIM**) as follows: for a given NP head word and a key word as listed in Table 2, the WordNet similarity package (Seco et al., 2004) models the length of path traveling from the head word to the key word over the WordNet network. It then computes the semantic similarity based on the path. For example, the similarity between *company* and *social group* is 0.77, while the similarity between *company* and *person* is 0.59. The key word which receives the highest similarity to the head word is marked as a feature.

The WN_CLASS feature may suffer from the coverage problem and the WN_SIM feature is heavily dependent on the definition of similarity metric which may turn out to be inappropriate for coreference resolution task. To make better use of WordNet knowledge, we attempt to directly introduce hypernyms for the NP head words (we denote

[2] It, however, is mostly useful for nominal noun phrase and not for the pronoun and proper noun phrases.

[3] Only the first synset of the NP is used.

it as **WN_HYP** feature). The most similar work to ours is (Daumé III and Marcu, 2005), in which two most common synsets from WordNet for *all* words in an NP and their hypernyms are extracted as features. We avoid augmenting the hypernyms for non-head words in the NP to prevent introducing noisy information, which may potentially corrupt the hypernym feature space.

Considering a WordNet hypernym structure as shown in Fig. 1 for the word *company*, its first synset (an institution created to conduct business) has a unique id of 08058098 and can also be represented by a set of description words (company in this case). Its third synset (the state of being with someone) has an id of 13929588 and description words of company, companionship, fellowship, society. Each synset can be extended by its hypernym synsets. For example, the direct hypernym of the first synset is the synset of 08053576 which can be described as institution, establishment. The augmentation of hypernyms for NP head words can introduce useful information, but can also bring noise if the head word or the synset of head word are not correctly identified. For an optimal use of WordNet hypernyms, four questions shall be addressed: 1) how many depths are required to tradeoff the generality (thus more informative) and the specificity (thus less noisy)? 2) which synset of the given word is needed to be augmented? 3) which representation (synset id or synset word) is better? and 4) is it helpful to encode the hypernym depth into the hypernym feature?[4] These four questions provide the guideline to search the optimal use of WordNet. We will design experiments in Section 3.5 to determine the optimal configuration of WN_HYP feature.

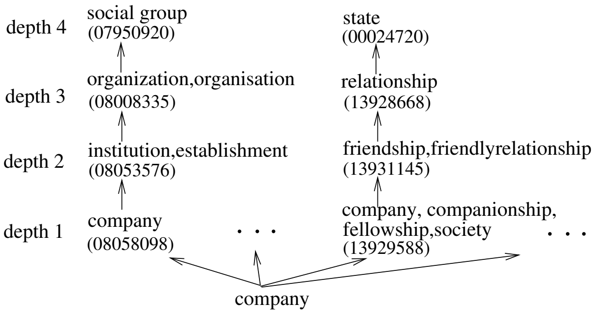

Figure 1: WordNet hypernym hierarchy for the word *company*.

3.4 Learning algorithm

Maximum entropy (ME) models (Berger et al., 1996; Manning and Klein, 2003), also known as log-linear and exponential learning models, has been adopted in the SC classification task. Maximum entropy models can integrate features from many heterogeneous information sources for classification. Each feature corresponds to a constraint on the model. Given a training set of (C, D), where C is a set of class labels and D is a set of feature represented data points, the maximum entropy model attempts to maximize the log likelihood

$$\log P(C|D,\lambda) = \sum_{(c,d)\in(C,D)} \log \frac{\exp \sum_i \lambda_i f_i(c,d)}{\sum_{c'} \exp \sum_j \lambda_j f_i(c,d)} \quad (1)$$

where $f_i(c, d)$ are feature indicator functions and λ_i are the parameters to be estimated. We use ME models for both SC classification and mention pair classification.

3.5 SC classification evaluation

We design three experiments to test the accuracy of our classifiers. The first experiment evaluates the individual contribution of different feature sets to SC classification accuracy. In particular, a ME model is trained on the 55,629 training instances using the following feature sets separately: 1) unigram, 2) bigram, 3) first-last word, 4) head word (HW), 5) named entities (NE), 6) HW+WN_CLASS, 7) HW+WN_SIM, and 8) variants of HW+WN_HYP. Note that HW+WN_CLASS is the semantic feature used in (Ng, 2007a), HW+WN_SIM is the semantic feature using WordNet similarity measure (Seco et al., 2004), and variants of HW+WN_HYP are the work proposed in this paper. We combine head word and the semantic features due to the fact that WordNet features are dependent on head words and they could be treated as units. In the second experiment, features are fed into the ME model incrementally until all features have been used.[5] Finally, we perform the feature ablation experiments. That is, we remove one feature at a time from the entire feature set and test the accuracy loss. The SC classification performance is measured by accuracy, i.e., the proportion of the correctly classified instances among all test instances.

Individual feature contribution Table 3 shows the SC classification accuracy of all NPs (all) and non-pronoun NPs (non-PN) on the development and test datasets using individual feature

[4] For example, we encode the synset 08053576 as 08053576-1, with the last digit 1 indicating the depth of hypernym with regard to the entry word *company*.

[5] The optimal of HW+WN_HYP configuration is used.

sets. Among all the lexical features, unigram feature performs the best (81.3%) for all NPs over the development dataset. The bigram feature performs poorly due to the sparsity problem: NPs usually consist of one to three words. The first-last word feature effectively models the prefix words (such as *a* and *the*) and the head words and thus obtains a reasonably high accuracy of 80.1%. As mentioned before, the head word feature may suffer from the sparsity and it results in the accuracy of 78.2%. We also list the accuracies for non-pronoun NP SC classification, which are slightly different compared to all NP SC classification except for bigram, in which the accuracy has increased 3.9%.

Table 3: SC classification accuracy of ME using individual feature sets for development and test ACE2 datasets.

Feature type	dev		test	
	all	non-PN	all	non-PN
unigram	81.3	81.6	72.4	71.9
bigram	32.5	36.4	26.3	28.4
first-last word	80.1	80.2	71.6	71.0
HW	78.2	78.0	68.3	67.1
NE	74.0	82.8	73.1	81.9
HW+WN_CLASS	79.5	79.4	70.3	69.5
HW+WN_SIM	81.2	81.4	73.8	73.6
HW+WN_HYP (1)	82.6	83.1	74.8	74.7
HW+WN_HYP (3)	82.8	83.4	75.2	75.2
HW+WN_HYP (6)	**83.1**	83.7	75.6	75.7
HW+WN_HYP (9)	83.0	83.6	75.7	75.7
HW+WN_HYP (∞)	83.1	83.7	75.8	75.9
HW+WN_HYP (6) word form	82.8	83.3	75.6	75.7
HW+WN_HYP (6) depth encoded	82.9	83.5	75.4	75.4
HW+WN_HYP (6) first synset	83.0	83.6	76.4	76.6

Although Stanford NER performs well on named entity recognition task, it results in accuracy of 74.0% for all NP SC classification, due to its inability to deal with pronouns such as *he* and common nouns such as *the government*. The removal of pronouns significantly boosts its accuracy to 82.8%. The introduction of semantic feature HW+WN_CLASS boosts the performance to 79.5% compared to the head word alone of 78.2%. This conforms to (Ng, 2007a) that only small gain can be achieved using WN_CLASS feature. The HW+WN_SIM feature outperforms HW+WN_CLASS and the accuracy reaches 81.2%. For the variants of HW+WN_HYP, we first search the optimal depth. This is performed by using all synsets for NP head word, encoding the feature using synset id (rather than synset word), and no hypernym depth is encoded in the features. We try various depths of 1, 3, 6, 9 and ∞, with ∞ signifies that no depth constraint is imposed. The optimal depth of 6 is obtained with the accuracy of 83.1% over the development dataset. We then fix the depth of 6 to try using synset word as features, using synset id with depth encoded as features, and using first synset only. The results show that the optimum is to encode the features using hypernym synset id without hypernym depth information and all synsets are considered for hypernym extraction. This is slightly different from the previous finding (Soon et al., 2001; Lin, 1998b) that a coreference resolution system employing only the first WordNet synset performs slightly better than that employing more than one synset.[6] The best result reaches the accuracy of 83.1%. Although the best semantic feature only outperforms the best lexical feature by 1.8% on the development dataset, its gain in the test dataset is more significant (3.2%, from 72.4% to 75.6%).

Incremental feature contribution Once we use the training and development datasets to find the optimal configuration of HW+WN_HYP semantic feature, we use all lexical features and the optimal HW+WN_HYP feature incrementally to train an ME model over the combination of training and development datasets. Table 4 shows the SC classification accuracy of all NPs (all) and non-pronoun NPs (non-PN) on the training+development (we refer it as training hereafter) and test datasets.

Table 4: SC classification accuracy of ME using incremental feature sets for training and test ACE2 datasets.

Feature type	train		test	
	all	non-PN	all	non-PN
HW	87.8	89.0	68.6	67.6
+WN_HYP	87.8	89.0	75.7	75.8
+unigram	91.5	93.3	77.7	78.1
+bigram	93.1	95.2	78.7	79.2
+first-last word	93.2	95.3	78.8	79.3
+NE	93.4	95.6	**83.1**	**84.4**
Ng 2007a	-	85.0	-	83.3

Note that the significant higher accuracies in training compared to test are due to the overfitting problem. The interesting evaluation thus remains on the test data. As can be seen, the inclusion of more features results in higher performance. This is more obvious in the test dataset than in the training dataset. The inclusion of the

[6] In fact, the accuracy of the test data supports their claims. The accuracy using the first synset compared to using all synsets results in the accuracy increase from 75.6% to 76.4% for all NPs over the test dataset.

optimized WN_HYP feature (ie, using all synsets' hypernyms up to 6 depth and with synset id encoding) results in 7.1% increase for all NP SC classification over test data. This shows the effectiveness of the WN_HYP features to overcome the sparsity of head word feature. The unigram, bigram and first-last word features offer reasonable accuracy gain, and the final inclusion of NE boosts the overall performance to 83.1% for all NP and 84.4% for non pronoun NPs over test data. This result can be directly compared to the SC classification accuracy as reported in (Ng, 2007a), in which the highest accuracy is 83.3% for non pronoun NPs.[7] The large difference between the highest training accuracies is due to that our classifier is trained directly on the ACE2 training dataset, while their SC classifier was trained on BBN Entity Type Corpus (Weischedel and Brunstein, 2005), which is five times larger than the ACE2 corpus used by us. In addition to WordNet, they have adopted multiple knowledge sources which include BBN's IdentiFinder (this is equivalent to the Stanford NER in our work), BLLIP corpus and Reuters Corpus,[8] and dependency based thesaurus (Lin, 1998a). It is remarkable that our SC classifier can achieve even higher accuracy only using WordNet hypernym and NE features. It is worth noting that the small accuracy gain is indeed hard to achieve considering that the test data size is large (15360).

Feature ablation experiment We now perform the feature ablation experiments to further determine the importance of individual features. We remove one feature at a time from the entire feature set. Table 5 shows the SC classification accuracy of all NPs (all) and non-pronoun NPs (non-PN) on the training and test datasets respectively.

Table 5: SC classification accuracy of ME by removing one feature at a time for training and test ACE2 datasets.

Feature type	train		test	
	all	non-PN	all	non-PN
overall	93.4	95.6	83.1	84.4
-HW	93.4	95.5	82.9	84.2
-WN_HYP	93.4	95.5	82.6	83.8
-HW+WN_HYP	93.4	95.5	82.3	83.5
-unigram	93.4	95.5	82.9	84.2
-bigram	92.5	94.5	82.7	84.0
-first-last word	93.4	95.5	82.9	84.1
-NE	93.2	95.3	78.8	79.3

Again, the significant higher accuracies in training compared to test are due to overfitting. The removal of NE feature results in the largest accuracy loss of 4.3% (from 83.1% to 78.8%) for all nouns on test data. It follows WN_HYP (0.5% loss) and the bigram (0.4%). If we treat HW+WN_HYP as one feature, the removal of it results in accuracy loss of 0.8% for all nouns on test data. The unigram, first-last word and head word each results in the loss of 0.2%. The reason that the removal of NE results in a much significant loss is due to the fact that the NE feature is quite different from other features. Its strength is to distinguish SCs for proper names, while other features are more similar (their targets are common nouns). The proposed use of HW+WN_HYP can bring 0.8% gain on top of other features, higher than other informative lexical features including unigram and first-last word.

3.6 Error analysis

A closer look at the errors produced by our SC classifier reveals that the second probable label is very likely to be the actual labels if the first probable one is wrong. In fact, if we allow the classifier to predict two most probable labels and the classification is judged to be true if the actual label is one of the two predictions, then the classification accuracy increases from 83.1% to 96.4%. This is because that the same noun phrases are sometimes annotated with one of the five explicitly annotated classes while sometimes are not annotated at all (thus falling into the other SC). Again for the example of *people*. It is annotated as *person* SC 20 times and is not annotated at all 21 times. Given the same feature set for this instance, the best the classifier can do is to classify it to *other* semantic class. To address this annotation inconsistency issue, we relax the SC agreement feature from the strict match in designing coreference resolution features. For example, if the first probable SC of an NP matches the second probable SC of another NP, we still give some partial match credit.

4 Application to Coreference Resolution

We can now incorporate the NP SC classifier into our ME based coreference resolution system. This section examines how our WordNet hypernym features help improve the coreference resolution performance.

4.1 Experimental setup

We use the ACE-2 (version 1.0) coreference corpus. Each raw text in this corpus was preprocessed automatically by a pipeline of NLP components, including sentence boundary detection,

[7] All NP accuracy was not reported as they excluded the pronouns in creating their training and test data.

[8] They use these corpus to extract patterns to induce SC of common nouns.

POS-tagging and text chunking. The statistics of corpus and mention extraction are shown in Table 6, where g-mention is the automatically extracted mentions which contain the annotated (gold) mentions. The recalls of gold mentions are 95.88% and 95.93% for training and test data respectively.

Table 6: Statistics for corpus and extracted mentions.

	text#	mention#	g-mention#	gold#	recall(%)
train	422	61810	22990	23977	95.88
test	97	15360	5561	5797	95.93

Our coreference system uses Maximum Entropy model to determine whether two NPs are coreferent. As in (Soon et al., 2001; Ponzetto and Strube, 2006), we generate training instances as follows: a positive instance is created for each anaphoric NP, NP_j, and its closest antecedent, NP_i; and a negative instance is created for NP_j paired with each of the intervening NPs, NP_{i+1}, NP_{i+2}, ..., NP_{j-1}. Each instance is represented by syntactic or semantic features described as follows. All training data are used to train a maximum entropy model. In the test stage,we select the closest preceding NP that is classified as coreferent with NP_j as the antecedent of NP_j. If no such NP exists, no antecedent is selected for NP_j.

Unlike other natural language processing tasks such as information extraction which have de facto evaluation metrics, it is an open question which evaluation is the most suitable one. The evaluation becomes more complicated when automatically extracted mentions (in contrast to the gold mentions) are used. To facilitate the comparison with previous work, we report performance using two different scoring metrics: the commonly-used MUC scorer (Vilain et al., 1995) and the *accuracy* of the anaphoric references (Ponzetto and Strube, 2006). An anaphoric reference is correctly resolved if it and its closest antecedent are in the same coreference chain in the resulting partition.

4.2 Baseline features

We briefly review the baseline features used in this paper as follows. More detailed information and implementations can be found at (Soon et al., 2001; Versley et al., 2008). For example, the ALIAS feature takes values of true or false. The value of true means that the antecedent and the anaphor refer to the same entity (date, person, organization or location). The ALIAS feature detection works differently depending on the named entity type. For date, the day, month, and year values are extracted and compared. For person, the last words of the noun phrases are compared. For organization names, the alias detection checks for acronym match such as *IBM* and *International Business Machines Corp*.

Lexical features STRING MATCH: true if NP_i and NP_j have the same spelling after removing article and demonstrative pronouns, false otherwise. ALIAS: true if NP_j is the alias of NP_i.

Grammatical features I_PRONOUN: true if NP_i is a pronoun; J_PRONOUN: true if NP_j is pronoun; J_REFL_PRONOUN: true if NP_j is reflexive pronoun; J_PERS_PRONOUN: true if NP_j is personal pronoun; J_POSS_PRONOUN: true if NP_j is possessive pronoun; J_PN: true if NP_j is proper noun; J_DEF: true if NP_j starts with *the*; J_DEM: true if NP_j starts with *this*, *that*, *these* or *those*; J_DEM_NOMINAL: true if NP_j is a demonstrative nominal noun; J_DEM_PRONOUN: true if NP_j is a demonstrative pronoun; PROPER_NAME: true if both NP_i and NP_j are proper names; NUMBER: true if NP_i and NP_j agree in number; GENDER: true if NP_i and NP_j agree in gender; APPOSITIVE: true if NP_i and NP_j are appositions.

Distance feature DISTANCE: how many sentences NP_i and NP_j are apart.

Semantic feature SEMCLASS: This feature is implemented from (Soon et al., 2001). Its possible values are true, false, or unknown. First the following semantic classes are defined: *female*, *male*, *person*, *organization*, *location*, *date*, *time*, *money*, *percent*, and *object*. Each of these defined semantic classes is then mapped to a WordNet synset. Then the semantic class determination module determines the semantic class for every NP as the first synset of the head noun of the NP. If such synset is a hyponym of defined semantic class, then such semantic class is assigned to the NP. Otherwise, *unknown* class is assigned. Finally, the agreement of semantic classes of NP_i and NP_j is unknown if either assigned class is *unknown*; true if their assigned class are the same, false otherwise. Notice that the WordNet use in (Ng, 2007a) and this feature apply in the same principle except that 1) the former is used in SC classification while the latter is used directly for coreference resolution, and 2) they have different semantic class categories.

4.3 Proposed WordNet agreement features

For each instance which consists of NP_i and NP_j, we apply our SC classifier to label them, say l_i and l_j respectively. We then use these two induced labels to propose the SC agreement feature for NP_i and NP_j. In particular, SC_STRICT is true if l_i and l_j are the same and they are not of other type, false otherwise; SC_COARSE is true if both l_i and l_j are not of other type; In addition, we propose two other SC agreement features to cope with the SC classification errors. SC_RELAX1 is true if the first probable of NP_i, l_{i1}, is not other type and is the same as the second probable of N_j, l_{j2}, or vice visa. SC_RELAX2 is true if the second probable of NP_i, l_{i2}, is not other type and is the same as the second probable label of NP_j, l_{j2}. The purpose in using SC_RELAX1 and SC_RELAX2 features is to relax the strict SC agreement feature in the hope that partial SC match is useful for coreference resolution.

4.4 Coreference results

Table 7 shows the MUC score for ACE2 corpus and its three partitions: bnews, npaper, and nwire using baseline and the proposed semantic features. It also shows the accuracy of resolving anaphors for all nouns in ACE2 corpus. SC_STRICT is the configuration that uses the baseline features with the SEMCLASS (Soon et al., 2001) replaced by SC_STRICT, and SC_COARSE, SC_RELAX1, and SC_RELAX2 are incrementally included into the SC_STRICT feature set.

As can be seen, the SC_STRICT significantly boosts the performance: it improves the MUC F score and anaphor accuracy of baseline from 57.7% to 65.7% and 37.7% to 46.3% respectively. It is remarkable that the new use of WordNet can obtain such significant gain in both MUC score and anaphor accuracy. The large improvement of the precision from 58.1% to 73.3% for all NPs shows that the SC_STRICT feature can effectively filter out the semantic incompatible pairs of antecedents and anaphors. In accordance with our hypothesis, the relaxation of strict SC agreement by including SC_COARSE, SC_RELAX1 and SC_RELAX2 help improve the performance further, which is reflected by both MUC score and anaphor accuracy. For example, compared to the baseline, the use of all proposed four SC agreement features results in the maximal accuracy gain of 9.7% (from 37.7% to 47.4%) and the use of SC_STRICT, SC_COARSE, and SC_RELAX1 results in the maximal MUC score gain of 10.4% (from 57.7% to 68.1%).

Our best MUC score is 68.1% which outperforms the MUC score of 64.6% as reported in (Ng, 2007a) by 3.5%, while our best accuracy of anaphor is 47.4%, which is 4.1% less than the accuracy of 51.5% in (Ng, 2007a). Note that, unlike (Ng, 2007a) which performed extensive experiments using different machine learning algorithms, alternative use of features (either constraint or normal features), and heterogeneous knowledge sources, this paper simply uses one learning classifier (ME model) and only employs WordNet and Stanford NER semantic sources.

The different MUC and accuracy scores reflect the non-trivial cases of evaluating coreference systems. While we leave out the discussion of which evaluation is more appropriate, we focus on showing that the proposed SC classifier can bring significant boost from the baseline using both MUC and accuracy metrics.

5 Conclusion

We have showed that the traditional use of WordNet in coreference resolution may not effectively exploit the WordNet semantic knowledge. We proposed new ways to extract WordNet feature. This feature, along with other features such as named entity feature, can be used to build an accurate *semantic class* (SC) classifier. In addition, we analyzed the classification errors of the SC classifier and relaxed SC agreement features to cope with part of the classification errors. The proposed accurate SC classifier and the relaxation of SC agreement features can boost our baseline coreference resolution system by 10.4% and 9.7% using MUC score and anaphor accuracy respectively.

Acknowledgments

We wish to thank Yannick Versley for his support with BART coreference resolution system and the three anonymous reviewers for their invaluable comments. This research was supported by British Telecom grant CT1080028046 and BISC Program of UC Berkeley.

References

A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

M. Collins 1999. Head-driven statistical models for natural language parsing. *PhD thesis*, University of Pennsylvania.

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

Table 7: MUC score and accuracy of baseline and proposed SC agreement features for ACE2 dataset.

	MUC score												Accuracy
	All			bnews			npaper			nwire			All
System	R	P	F	R	P	F	R	P	F	R	P	F	
baseline	57.4	58.1	57.7	56.6	55.4	56.0	59.3	60.4	59.9	56.2	58.6	57.3	37.7
Ng 2007a	59.5	70.6	64.6	-	-	-	-	-	-	-	-	-	**51.5**
SC_STRICT	59.6	73.3	65.7	61.6	72.8	66.7	60.3	74.9	66.8	56.8	72.1	63.5	46.3
+ SC_COARSE	59.2	76.7	66.8	61.0	76.7	67.9	59.8	77.2	67.4	56.6	76.2	64.9	45.9
+ SC_RELAX1	59.8	79.0	**68.1**	61.3	79.8	69.3	60.9	80.3	69.3	57.2	76.7	65.5	47.2
+ SC_RELAX2	60.2	77.7	67.8	61.5	78.2	68.9	61.4	78.9	69.1	57.5	75.7	65.4	47.4

P. Denis and J. Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *HLT-NAACL*.

C. Fellbaum. 1998. *An electronic lexical database*. The MIT press.

J. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. of ACL*, pages 363-370.

R. Florian, H. Jing, N. Kambhatla, and I. Zitouni. 2006. Factorizing complex models: a case study in mention detection. In *Proc. of COLING/ACL*, pages 473-480.

S. M. Harabagiu, R. C. Bunescu, and S. J. Maiorano. 2001. Text and knowledge mining for coreference resolution. In *Proc. of NAACL*, pages 55-62.

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

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

D. Lin. 1998b. Using collocation statistics in information extraction. In *Proc. of MUC-7*.

X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. 2004. A mention synchronous coreference resolution algorithm based on the Bell tree. In *Proc. of the ACL*.

C. Manning and D. Klein. 2003. Optimization, Maxent Models, and Conditional Estimation without Magic. *Tutorial at HLT-NAACL 2003 and ACL 2003*.

K. Markert and M. Nissim. 2005. Comparing knowledge sources for nominal anaphora resolution. *Computational Linguistics*, 31(3):367-401.

A. McCallum and B. Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Proc. of the NIPS*.

V. Ng. 2007a. Semantic Class Induction and Coreference Resolution. In *Proc. of the ACL*.

V. Ng. 2007b. Shallow Semantics for Coreference Resolution. In *Proc. of the IJCAI*.

C. Nicolae and G. Nicolae 2006. BESTCUT: A Graph Algorithm for Coreference Resolution. In *Proc. of the EMNLP*.

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

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

N. Seco, T. Veale, and J. Hayes. 2004. An Intrinsic Information Content Metric for Semantic Similarity in WordNet. *Proc. of the European Conference of Artificial Intelligence*.

W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computation Linguistics*, 27(4):521-544.

Y. Versley, S. P. Ponzetto, M. Poesio, V. Eidelman, A. Jern, J. Smith, X. Yang, and A. Moschitti. 2008. BART: a modular toolkit for coreference resolution. *ACL 2008 System demo*.

R. Vieira and M. Poesio. 2000. An empirically-based system for processing definite descriptions. *Computational Linguistics*, 26(4):539-593.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoreing scheme. In *Proc. of MUC-6*, pages 45-52.

R. Weischedel and A. Brunstein. 2005. BBN pronoun coreference and entity type corpus. Linguistic Data Consortium.

X. Yang and J. Su. 2007. Coreference resolution using semantic relatedness information from automatically discovered pattens. In *Proc. of the ACL*.