Supervised Ranking for Pronoun Resolution: Some Recent Improvements

Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
vince@hlt.utdallas.edu

Abstract

A recently-proposed machine learning approach to reference resolution — the *twin-candidate* approach — has been shown to be more promising than the traditional *single-candidate* approach. This paper presents a pronoun interpretation system that extends the twin-candidate framework by (1) equipping it with the ability to identify non-referential pronouns, (2) training different models for handling different types of pronouns, and (3) incorporating linguistic knowledge sources that are generally not employed in traditional pronoun resolvers. The resulting system, when evaluated on a standard coreference corpus, outperforms not only the original twin-candidate approach but also a state-of-the-art pronoun resolver.

Introduction

While the majority of traditional pronoun resolvers are knowledge-based systems, machine learning approaches to pronoun interpretation and the broader task of noun phrase (NP) coreference have become increasingly popular in recent years (see Mitkov (2002) for an overview). Learning-based pronoun resolution systems operate primarily by (1) training a *coreference* classifier that determines whether (or how likely) a pronoun and a candidate antecedent are coreferring, and then (2) resolving a pronoun to its closest (or in some cases the most probable) preceding coreferent NP (e.g., Soon *et al.* (2001), Kehler *et al.* (2004)).

An alternative to the above *single-candidate* (SC) learning approach to reference resolution is the *twin-candidate* (TC) ranking approach proposed independently by Iida *et al.* (2003) and Yang *et al.* (2003). In the TC approach, a *preference* classifier is trained that, given a pronoun and two of its candidate antecedents, determines which of the two is more likely to be the antecedent of the pronoun. A separate mechanism then coordinates these preference decisions and selects the most preferred candidate to be the antecedent.

One appeal of the TC approach lies in its potentially stronger modeling capability than the SC approach: the fact that the TC classifier is conditioned on a pronoun and *two* of its candidate antecedents (as opposed to one in SC) makes it possible to model the relationship between the two candidates. Indeed, empirical results have demonstrated the superiority of the TC approach to its SC counterpart.

Our goal in this paper is to improve pronoun resolution by investigating three modifications to the TC approach:

Identifying non-referential pronouns. Apart from truly non-referential pronouns (e.g., pleonastic pronouns), a system is often not supposed to resolve several types of referential pronouns (e.g., pronouns that refer to clausal constructions) when evaluated on standard coreference corpora such as MUC and ACE. Hence, it is imperative for a resolver to identify these “non-referential” pronouns for which no antecedent should be selected. However, as we will see, the TC approach lacks a mechanism for classifying a pronoun as non-referential. We will remedy this deficiency by proposing a new method for detecting non-referential pronouns that can be incorporated naturally into the TC framework.

Training specialized classifiers. Different types of pronouns have different linguistic properties (e.g., reflexives vs. non-reflexives, singular vs. plural) and hence may require different resolution strategies. While virtually all learning-based pronoun resolvers simply acquire one model to handle all types of pronouns, training one model for each pronoun type can potentially allow a learner to better acquire different resolution strategies for different pronoun types. We will investigate the latter possibility in this paper.

Employing contextual knowledge. Many existing pronoun resolvers operate by relying on a set of morphosyntactic cues. Kehler *et al.* (2004) observe that the performance of these systems is plateauing, speculating that further progress in pronoun interpretation would require the use of deeper linguistic knowledge. Hence, we will explore the possibility of exploiting *contextual* knowledge (i.e., the context in which a pronoun and its candidate antecedents occur).

We evaluate our extended TC approach against two baseline systems, one adopting the SC approach and the other the TC approach (without our modifications). Experimental results on the ACE coreference corpus demonstrate the effectiveness of our approach in improving both baselines. Furthermore, our system’s accuracy of 80.5% on referential pronoun resolution compares favorably to a state-of-the-art pronoun resolver developed by Kehler *et al.* (2004), which achieves an accuracy of 76.8% on the same corpus.

The rest of the paper is organized as follows. After discussing related work, we describe the two baseline systems in detail. We then elaborate on our three modifications to the TC framework and present evaluation results.

Related Work

In this section, we will center the discussion of related work on pronoun resolution around the three modifications that we outlined in the introduction.

Identifying non-referential pronouns. Many classic and recent pronoun resolution systems (e.g., Hobbs (1978), Brennan *et al.* (1987), Strube (1998), Tetreault (2001)) focus on the resolution of referential pronouns and simply ignore the problem of recognizing non-referential pronouns. Nevertheless, there is a large body of work that aims at identifying specific types of non-referential phrases such as pleonastic pronouns (e.g., Lappin and Leass (1994), Kennedy and Boguraev (1996)) and non-referential definite descriptions (e.g., Bean and Riloff (1999), Vieira and Poesio (2000)). There have also been some attempts on augmenting a reference resolver with a pre-processing module for identifying and filtering non-referential phrases (e.g., Byron and Gegg-Harrison (2004), Ng and Cardie (2002)). In contrast to previous work, we will show how we can integrate non-referential pronoun recognition with referential pronoun resolution within the TC learning framework, thereby obviating the need to employ a separate pre-processing module.

Training specialized classifiers. We are not aware of any learning-based approaches to pronoun resolution that attempt to train different classifiers for handling different types of pronouns. One plausible reason is that research in learning-based pronoun resolution (e.g., Ge *et al.* (1998), Kehler *et al.* (2004)) has focused largely on feature development and may have overlooked the relevant machine learning issues. Another reason may be that learning-based pronoun resolution is often studied in the context of NP coreference (e.g., Aone and Bennett (1995), McCarthy and Lehnert (1995), Soon *et al.* (2001)), in which researchers expend their efforts on issues surrounding the broader coreference task that may be less critical for pronoun resolution.

Employing contextual knowledge. There have been several attempts on employing deeper linguistic knowledge than that provided by morphosyntactic features. For instance, Soon *et al.* (2001) employ a named entity recognizer and the WordNet semantic knowledge base to determine the semantic class of an NP. Iida *et al.* (2003) explore the use of discourse-level features motivated by the centering theory. Kehler *et al.* (2004) extend a method originally proposed by Dagan and Itai (1990) for learning selectional regularities (a form of shallow semantic knowledge) and applying such knowledge to resolving pronoun references. Bean and Riloff (2004) present an unsupervised method for acquiring contextual knowledge using extraction patterns. Motivated in part by Kehler *et al.* and Bean and Riloff, we will explore a different form of contextual knowledge in this paper.

Baseline Pronoun Resolution Systems

In this section, we will describe our implementation of two baseline reference resolvers: the Soon *et al.* (2001) system, which employs the SC approach, and the Yang *et al.* (2003) system, which employs the TC approach. As we will see, both systems will be trained on data annotated with coreference chains. Since all elements preceding a pronoun in a given coreference chain are correct antecedents of the pronoun, successfully resolving the pronoun amounts to selecting one of these preceding elements as the antecedent.

Soon *et al.*'s Single-Candidate Approach

Training the model. We train a *coreference* classifier that, given a description of a pronoun, NP$_k$, and one of its preceding NPs, NP$_j$, decides whether or not they are co-referring. Thus, each training instance represents two noun phrases, NP$_j$ and NP$_k$. The classification associated with a training instance is one of POSITIVE or NEGATIVE depending on whether the two NPs co-refer in the associated training text.

We follow the procedure employed in Soon *et al.* to create training instances: we rely on coreference chains from the answer keys to create (1) a *positive instance* for each referential pronoun, NP$_k$, and its closest antecedent, NP$_j$; and (2) a *negative instance* for NP$_k$ paired with each of the intervening NPs, NP$_{j+1}$, NP$_{j+2}$,. . ., NP$_{k-1}$.

Applying the model. After training, the classifier is used to guide the selection of an antecedent for each pronoun in a test text. Specifically, each pronoun, NP$_k$, is compared in turn to each preceding NP, NP$_j$, from right to left. For each pair, a test instance is created as during training and is presented to the coreference classifier, which returns a number between 0 and 1 that indicates the likelihood that the two NPs are coreferent. NP pairs with class values above 0.5 are considered coreferent; otherwise the pair is considered not coreferent. The process terminates as soon as an antecedent for NP$_k$ or the beginning of the text is reached. In other words, NP$_k$ will be considered non-referential if none of its preceding NPs is coreferent with it.

Yang *et al.*'s Twin-Candidate Approach

Training the model. We train a *preference* classifier that, given a description of a pronoun, NP$_k$, and two of its candidate antecedents, NP$_i$ and NP$_j$, decides whether NP$_i$ or NP$_j$ is the preferred antecedent of NP$_k$. Hence, each training instance corresponds to three NPs, NP$_i$, NP$_j$, and NP$_k$. (Without loss of generality, we will assume that NP$_k$ is closer to NP$_j$ than to NP$_i$.) The classification associated with a training instance is one of POSITIVE (if NP$_j$ is the preferred antecedent) or NEGATIVE (if NP$_i$ is the preferred antecedent).

We follow the procedure employed in Yang *et al.* to generate training instances, relying on coreference chains from the answer keys to create (1) a *positive instance* from NP$_i$, NP$_j$, and NP$_k$ if NP$_j$ is coreferent with NP$_k$ but NP$_i$ is not; and (2) a *negative instance* from the three NPs if NP$_i$ is coreferent with NP$_k$ but NP$_j$ is not. In other words, a training instance is generated if and only if exactly one of the two candidates is the correct antecedent of NP$_k$. If this condition is not met, then neither of the candidates is preferable to the other and hence no training instances should be generated.

Note that the number of training instances created by this method is cubic in the number of NPs in the associated training text if we define the candidate set of a pronoun to be the set of all NPs preceding it. Hence, to reduce the training time, we reduce the number of training instances by restricting the candidate set of a pronoun to contain only the preceding NPs in either the same sentence as the pronoun or any of the immediately preceding three sentences.

Features describing a candidate antecedent		
1	PRONOUN_1	1 if NP$_i$/NP$_j$ is a pronoun; else 0.
2	PROPER_NOUN_1	1 if NP$_i$/NP$_j$ is a proper noun; else 0.
3	DEFINITE_1	1 if NP$_i$/NP$_j$ is a definite NP; else 0.
4	INDEFINITE_1	1 if NP$_i$/NP$_j$ is an indefinite NP; else 0.
5	GRAM_ROLE_1	the grammatical role of NP$_i$/NP$_j$ as extracted by Lin's (1998) MINIPAR dependency parser.
6	NAMED_ENTITY_1	1 if NP$_i$/NP$_j$ is a person; 2 if organization; 3 if location; else 0.
7	SEMCLASS_1	the WordNet semantic class of NP$_i$/NP$_j$.[1]
Features describing the pronoun to be resolved		
8	GRAM_ROLE_2	the grammatical role of NP$_k$ as extracted by Lin's (1998) MINIPAR dependency parser.
9	CASE_2	1 if NP$_k$ has nominative case; 2 if accusative; 3 if possessive; 4 if reflexive; else 0.
10	NUMBER_2	1 if NP$_k$ is singular; 2 if plural; 0 if the number cannot be determined.
11	PERSON_2	if NP$_k$ is a first-person pronoun; 2 if second-person; 3 if third-person; 0 if the person cannot be determined.
12	STRING_2	the surface string of NP$_k$.
13	PRO_EQUIV_2	1 if there exists a preceding pronoun that is the same string as NP$_k$ or differs from it only w.r.t. case; else 0.
Features describing the relationship between a candidate antecedent and the pronoun to be resolved		
14	NUMBER	1 if the NPs agree in number; 0 if they disagree; 2 if the number for one or both NPs cannot be determined.
15	GENDER	1 if the NPs agree in gender; 0 if they disagree; 2 if the gender for one or both NPs cannot be determined.
16	SEMCLASS	1 if the NPs have the same WordNet semantic class; 0 if they don't; 2 if the semantic class information for one or both NPs cannot be determined.
17	PRO_STR	1 if both NPs are pronominal and are the same string; else 0.
18	PRO_EQUIV	1 if the NPs are the same pronoun (but may differ w.r.t. case); else 0.
19	BOTH_SUBJECTS	1 if both NPs are grammatical subjects; 0 if neither are subjects; else 2.
20	AGREEMENT	1 if the NPs agree in both gender and number; 0 if they disagree in both gender and number; else 2.
21	PARANUM	distance between the NPs in terms of the number of paragraphs.
22	SENTNUM	distance between the NPs in terms of the number of sentences.
23	GRAM_ROLE	1 if the NPs have the same grammatical role; else 0.
24	STR_CONCAT	the concatenation of the strings of the two NPs.

Table 1: Feature Set for the Baseline Systems.

Applying the model. After training, the learned classifier is used to guide the selection of an antecedent for each pronoun in a test text. For efficiency reasons, the three-sentence window employed in training is also used to limit the size of the candidate set for each pronoun in the test set. To select an antecedent for NP$_k$, we first initialize the score of each candidate antecedent to zero. Next, for each pair of candidate antecedents, NP$_i$ and NP$_j$, we create a test instance involving NP$_i$, NP$_j$, and NP$_k$ as in training and present it to the classifier. If the classifier determines that NP$_j$ is preferable to NP$_i$, we increment the score of NP$_j$ by 1; otherwise, we increment the score of NP$_i$ by 1. Finally, the candidate with the highest score is selected to be the antecedent of NP$_k$. In case of ties, the highest-scored candidate that is closest to NP$_k$ is selected.

As can be seen, the TC approach lacks a mechanism for classifying a pronoun as non-referential. To address this problem, Yang *et al.* first apply the SC approach to identify and filter the non-referential pronouns, and then use the TC approach to resolve only those pronouns that survive the SC filter. We will explore a different approach to non-referential pronoun identification in the next section.

Remaining Implementation Issues

To implement these two approaches, we also need to specify (1) the *learning algorithm* used to train the classifiers and (2) the set of features used to represent an instance.

Learning algorithm. We use SVM^{light} (Joachims 1999), a publicly-available implementation of the support vector machine (SVM) learner, to train the classifiers.

Feature set. To build strong baseline classifiers, we designed a feature set that is composed of selected features employed by high-performing reference resolvers such as Soon *et al.* (2001), Kehler *et al.* (2004), and Yang *et al.* (2004).

The feature set (shown in Table 1) is composed of 24 features that can be divided into three types: (i) features describing a candidate antecedent; (ii) features describing the pronoun to be resolved; and (iii) features describing the relationship between them.[2] As can be seen, seven features are used to describe a candidate's NP type (e.g., whether it is a pronoun (PRONOUN_1) or a proper noun (PROPER_NOUN_1)), its definiteness (e.g., whether it is a definite NP (DEFINITE_1) or an indefinite NP (INDEFINITE_1)), and its grammatical role. Moreover, six features are used to characterize the lexical and grammatical properties of the pronoun to be resolved, including its surface string, grammatical role, case, number, and person. Finally, 11 features are used to describe the relationship between a candidate and the pronoun, checking for number, gender, and semantic class agreement, as well as measuring the distance between them in sentences and paragraphs.

In the SC approach, an instance involving NP$_j$ and NP$_k$ is

[1] The semantic classes we considered are: person, organization, time, day, money, percent, measure, abstraction, psychological feature, phenomenon, state, group, object, and unknown.

[2] For ease of exposition, some of the nominal features are presented as multi-class features. To represent these features on a binary scale for use by an SVM, we transform each multi-class nominal variable to an equivalent set of variables with binary values.

represented using these 24 features: the seven type (i) features are used to describe NP_j, the six type (ii) features are used to describe NP_k, and the 11 type (iii) features are used to describe their relationship. In the TC approach, on the other hand, an instance involving NP_i, NP_j, and NP_k is represented using a total of 42 features: in addition to the 24 features employed by the SC approach to characterize NP_j, NP_k, and the relationship between them, we also use the seven type (i) features to describe NP_i and the 11 type (iii) features to describe the relationship between NP_i and NP_k.

Three Modifications

This section details our three modifications to the learning framework underlying the TC approach.

Identifying non-referential pronouns. To equip the TC approach with the ability to identify non-referential pronouns, we augment the candidate set of each pronoun in the training and test texts with a *null* antecedent. The idea is that selecting *null* to be the antecedent of a pronoun amounts to classifying the pronoun as non-referential.

Now, to enable the learner to learn when *null* is preferable to NP_j as the antecedent of NP_k, additional training instances involving *null*, NP_j, and NP_k will be generated if (1) NP_k is non-referential, in which case *null* is the preferred antecedent; or (2) NP_j is an antecedent of NP_k, in which case NP_j is preferable to *null*. To represent an instance involving *null*, we employ essentially the same features as described in Table 1, except that no features will be generated for *null*, nor will there be features describing the relationship between *null* and NP_k. In addition, a new binary feature will be added to *all* training instances to indicate whether the corresponding instance involves a *null* candidate antecedent.

Testing proceeds as in Yang *et al.*'s system, except that *null* is now a candidate antecedent for each pronoun to be resolved (and hence additional test instances will be generated). A *null* candidate is scored in the same way as its non-null counterparts. As mentioned above, if *null* achieves the highest score among the candidate antecedents, the corresponding pronoun is assumed to be non-referential.

Training specialized classifiers. As mentioned in the introduction, training one classifier for each type of pronoun can potentially make the learning process easier by allowing a learner to better capture the linguistic properties specific to a given pronoun type. Now the question is: along which dimension should we split the pronouns?

We can, for instance, split them along the *number* dimension, thereby training one classifier for handling singular pronouns and another for plural pronouns; or we can split them along the *case* dimension and train one classifier for each of nominative, possessive, and accusative pronouns. To keep things simple, however, we choose to split the pronouns along a very fundamental dimension: *string*. Specifically, we train one classifier for classifying each set of pronouns that are lexically identical. Hence, one classifier will be trained on instances involving *he*; another one will be trained on instances involving *her*, for instance.

During testing, an instance will be classified by the respective specialized classifier (e.g., a *he* instance will be handled by the *he* classifier). However, instances for which the corresponding pronoun was not seen in the training data will be handled by the baseline classifier (i.e., the classifier learned from the full set of training instances). In essence, the baseline classifier serves as our “backoff model”.

	Newspaper		Newswire	
	Train	Test	Train	Test
Number of pronouns	2428	778	2756	611
% of annotated pronouns	79.8	78.4	76.2	73.5

Table 2: Statistics of the two ACE data sets

Employing contextual knowledge. We attempt to approximate the context in which an NP occurs using its *governor*. Roughly speaking, the governor of an NP is the lexical head of the node dominating the NP in the associated parse tree. For instance, in the PP *in a house*, the governor of *a house* is the preposition *in*; and in the VP *likes Mary*, the governor of *Mary* is the predicate *likes*. Hence, the governor of an NP can be thought of as a shallow representation of the context in which the NP appears. To incorporate governor information into our system, we expand the feature set to include a “governor” feature for each NP involved in an instance, with the feature value being the governor itself. In our experiments, we use Lin's (1998) MINIPAR dependency parser to compute the governor of an NP.

Evaluation

This section reports on the evaluation of our approach.

Experimental Setup

We use the newspaper (PA) and newswire (WI) segments of the Automatic Content Extraction (ACE) coreference corpus in our evaluation, training our classifiers on the training texts and evaluating our resolver on the test texts. Table 2 shows the number of personal and possessive pronouns in the training and test texts of each of the two data sets, as well as the percentage of these pronouns that are annotated. In ACE, a pronoun is annotated (or ACE-referential) only if it refers to an entity that belongs to one of the ACE named entity types. Hence, non-referential pronouns and pronouns referring to a non-ACE named entity are both unannotated (or non-ACE-referential). In the absence of complete coreference annotations in this corpus, we can only measure the performance of our approach with respect to the ACE-referential and non-ACE-referential pronouns.[3] Note, however, that as long as we are given a corpus in which all referential pronouns are annotated, our resolver can be easily trained to distinguish truly referential and truly non-referential pronouns.

Following the common practice of evaluating pronoun resolvers, we report performance in terms of accuracy. We adopt the standard notion of accuracy, defining (1) the accuracy of referential pronoun resolution to be the fraction of referential pronouns that are correctly resolved and (2) the

[3] For simplicity, we will use the term (*non-*)*referential pronouns* to refer to (non-)ACE-referential pronouns in our discussion of the ACE results, but the reader should bear in mind their differences.

		Newspaper (PA)				Newswire (WI)			
	System Variation	Referential		Non-referential		Referential		Non-referential	
1	Single-Candidate Baseline	419 / 610	(.6869)	41 / 168	(.2440)	332 / 449	(.7394)	43 / 162	(.2654)
2	Twin-Candidate Baseline (no filter)	447 / 610	(.7328)	0 / 168	(.0000)	337 / 449	(.7506)	0 / 162	(.0000)
3	Twin-Candidate Baseline (w/ filter)	444 / 610	(.7279)	41 / 168	(.2440)	324 / 449	(.7216)	43 / 162	(.2654)
4	Modified Learning Framework	467 / 610	(.7658)	**118 / 168**	**(.7024)**	**368 / 449**	**(.8196)**	**120 / 162**	**(.7407)**
5	null only	466 / 610	(.7639)	96 / 168	(.5714)	352 / 449	(.7840)	108 / 162	(.6667)
6	ensemble only	447 / 610	(.7328)	0 / 168	(.0000)	337 / 449	(.7506)	0 / 162	(.0000)
7	governor only	452 / 610	(.7410)	0 / 168	(.0000)	331 / 449	(.7372)	0 / 162	(.0000)
8	null + ensemble only	**470 / 610**	**(.7705)**	**118 / 168**	**(.7024)**	364 / 449	(.8107)	117 / 162	(.7222)
9	null + governor only	**470 / 610**	**(.7705)**	96 / 168	(.5714)	350 / 449	(.7795)	108 / 162	(.6667)
10	ensemble + governor only	450 / 610	(.7377)	0 / 168	(.0000)	357 / 449	(.7951)	0 / 162	(.0000)

Table 3: Results for the Newspaper and Newswire data sets. The accuracies of referential pronoun resolution and non-referential pronoun identification are shown. The best results obtained for a particular data set and pronoun type combination are boldfaced.

accuracy of non-referential pronoun identification to be the fraction of non-referential pronouns correctly identified.

Results and Discussion

Baseline systems. Results using the SC baseline are shown in row 1 of Table 3. In each column, accuracy is expressed first as a fraction and then as an equivalent decimal number. As we can see, the accuracy of referential pronoun resolution is .687 for PA and .739 for WI. Moreover, this baseline successfully identifies a number of non-referential pronouns, with an accuracy of .244 for PA and .265 for WI.

Rows 2 and 3 of Table 3 show the results using the TC baselines, which differ only in terms of whether the SC filter is applied to identify non-referential pronouns. Comparing rows 1 and 3, we see that the SC baseline and the "filtered" TC (FTC) baseline have the same accuracy of non-referential pronoun identification. This should not be surprising, since the two systems employ the same filter. Now, focusing on the two TC baselines, we note that the accuracy of referential pronoun resolution drops with the application of the SC filter for both data sets. In fact, the dramatic drop in accuracy for WI has caused the FTC baseline to perform worse than its SC counterpart. These results raise concerns regarding the robustness of the SC filtering method.

Modified learning framework. Results on our three modifications to the "unfiltered" TC (UTC) framework are shown in row 4 of Table 3. When used in combination, the modifications provide highly significant gains over the UTC baseline with respect to referential pronoun resolution[4], with accuracy increasing from .733 to .766 for PA and from .751 to .820 for WI. Equally encouraging is the fact that the accuracy of non-referential pronoun identification reaches 70-74%, which is almost triple that of the FTC baseline.

A closer look at the modifications. In an attempt to gain additional insight into the contribution of each modification to system performance, we apply each modification to the UTC baseline in isolation. Results are shown in rows 5-7 of Table 3. In comparison to the UTC baseline, we can see that training an *ensemble* of classifiers or incorporating the *governor*-related features alone only has a tiny impact on performance. On the other hand, the addition of *null* candidate antecedents to the UTC baseline yields large performance gains: about 57-67% of the non-referential pronouns are correctly identified. And perhaps more interestingly, although this modification is targeted at non-referential pronoun identification, we see significant improvement (p = .05) in referential pronoun resolution for both data sets: accuracy rises from .733 to .764 for PA and from .751 to .784 for WI. We speculate that referential pronoun resolution has benefited from the acquisition of a more accurate preference classifier as a result of the incorporation of the *null*-related training instances, but this remains to be verified.

To further our investigation of these modifications, we also apply *pairs* of modifications to the UTC baseline. Results are shown in rows 8-10 of Table 3. Perhaps not surprisingly, the addition of the *governor*-related features on top of *null* only yields a slight change in accuracy (compare rows 5 and 9), since these features do not seem to contribute much to performance, as described above. On the other hand, applying *null* and *ensemble* in combination yields better performance in both referential pronoun resolution and non-referential pronoun identification than applying *null* only (compare rows 5 and 8). This is somewhat surprising, since using *ensemble* alone is not effective at improving the UTC baseline (compare rows 2 and 6). These results imply that the modifications interact with each other in a non-trivial manner, suggesting that additional performance gains might be obtained by further investigating their interaction.

Comparison with Kehler *et al.*'s system. To get a better idea of how well our approach performs, we compare it with a state-of-the-art pronoun resolver developed by Kehler *et al.* (2004). We chose this system primarily because it was also evaluated on the ACE corpus. The system adopts an SC-like approach, employing a variety of morphosyntactic cues as features for their maximum entropy and Naive Bayes learners. However, unlike our system, their resolver only handles third-person referential pronouns.

We made a good-faith effort to duplicate the experimental conditions under which their system was evaluated. In particular, they trained their resolver on the combined PA/WI training set and evaluated it on the combined PA/WI test set, reporting an accuracy of .768 for referential pronoun resolution. When evaluated under the same condition, our system

[4] Chi-square statistical significance tests are applied to changes in accuracy, with p set to .01 unless otherwise stated.

		MUC-6		MUC-7	
	System Variation	Referential	Non-referential	Referential	Non-referential
1	Single-Candidate Baseline	253 / 363 (.6970)	14 / 54 (.2593)	111 / 220 (.5045)	**22 / 54 (.4074)**
2	Twin-Candidate Baseline (no filter)	265 / 363 (.7300)	0 / 54 (.0000)	126 / 220 (.5727)	0 / 54 (.0000)
3	Twin-Candidate Baseline (w/ filter)	259 / 363 (.7135)	14 / 54 (.2593)	114 / 220 (.5182)	**22 / 54 (.4074)**
4	Modified Learning Framework	**271 / 363 (.7466)**	**34 / 54 (.6296)**	**135 / 220 (.6136)**	20 / 54 (.3704)

Table 4: Results for the MUC-6 and MUC-7 data sets. The accuracies of referential pronoun resolution and non-referential pronoun identification are shown. The best results obtained for a particular data set and pronoun type combination are boldfaced.

yields an accuracy of .805, which represents a statistically significant improvement over Kehler *et al.*'s result.

Results on the MUC corpus. In an attempt to measure the performance of our resolver on a corpus annotated with complete NP coreference information, we repeat the above experiments on the MUC-6 and MUC-7 data sets. Results are shown in Table 4. Overall, the performance trends on the MUC data sets and the ACE data sets are similar. The only exception seems to be that our modifications do not improve the FTC baseline with respect to non-referential pronoun identification on the MUC-7 data set. While additional analysis is required to determine the reason, it is apparent that the SC filter employed by the FTC baseline is sacrificing the accuracy of referential pronoun resolution for that of non-referential pronoun identification in this case by removing pronouns overly liberally.

Conclusions

We have presented a pronoun resolution system that extends the twin-candidate learning framework with three modifications. Experiments on two ACE coreference data sets show that our system outperforms not only the original twin-candidate approach but also Kehler *et al.*'s pronoun resolver. Among the three modifications, the use of governor-related features has the least impact on performance. On the other hand, the introduction of *null* candidate antecedents not only enables us to handle the two tasks —referential pronoun resolution and non-referential pronoun identification —in a uniform manner within the twin-candidate framework, but also yields performance improvements on both tasks. Finally, training specialized classifiers does not improve the twin-candidate baseline when applied in isolation, but offers substantial gains when it is used in the presence of the *null* candidate antecedents. Although our resolver is still far from perfect, we believe that our work represents another step towards accurate pronoun resolution.

References

Aone, C., and Bennett, S. W. 1995. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proc. of the ACL*, 122–129.

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

Bean, D., and Riloff, E. 2004. Unsupervised learning of contextual role knowledge for coreference resolution. In *Proc. of HLT/NAACL*, 297–304.

Brennan, S. E.; Friedman, M. W.; and Pollard, C. J. 1987. A centering approach to pronouns. In *Proc. of the ACL*, 155–162.

Byron, D., and Gegg-Harrison, W. 2004. Eliminating non-referring noun phrases from coreference resolution. In *Proc. of DAARC*, 21–26.

Dagan, I., and Itai, A. 1990. Automatic processing of large corpora for the resolution of anaphora references. In *Proc. of COLING*, 330–332.

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

Hobbs, J. 1978. Resolving pronoun references. *Lingua* 44:311–338.

Iida, R.; Inui, K.; Takamura, H.; and Matsumoto, Y. 2003. Incorporating contextual cues in trainable models for coreference resolution. In *Proc. of the EACL Workshop on The Computational Treatment of Anaphora*.

Joachims, T. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press. 44–56.

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

Kennedy, C., and Boguraev, B. 1996. Anaphor for everyone: Pronominal anaphora resolution without a parser. In *Proc. of COLING*, 113–118.

Lappin, S., and Leass, H. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics* 20(4):535–562.

Lin, D. 1998. Dependency-based evaluation of MINIPAR. In *Proc. of the LREC Workshop on the Evaluation of Parsing Systems*, 48–56.

McCarthy, J., and Lehnert, W. 1995. Using decision trees for coreference resolution. In *Proc. of IJCAI*, 1050–1055.

Mitkov, R. 2002. *Anaphora Resolution*. Longman.

Ng, V., and Cardie, C. 2002. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proc. of COLING*, 730–736.

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

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

Strube, M. 1998. Never look back: An alternative to centering. In *Proc. of COLING-ACL*, 1251–1257.

Tetreault, J. 2001. A corpus-based evaluation of centering and pronoun resolution. *Computational Linguistics* 27(4):507–520.

Yang, X.; Zhou, G.; Su, J.; and Tan, C. L. 2003. Coreference resolution using competitive learning approach. In *Proc. of the ACL*, 176–183.

Yang, X.; Su, J.; Zhou, G.; and Tan, C. L. 2004. Improving pronoun resolution by incorporating coreferential information of candidates. In *Proc. of the ACL*, 128–135.