A Twin-Candidate Model for Learning-Based Anaphora Resolution

Xiaofeng Yang[*]
Institute for Infocomm Research

Jian Su[**]
Institute for Infocomm Research

Chew Lim Tan[†]
School of Computing,
National University of Singapore

The traditional single-candidate learning model for anaphora resolution considers the antecedent candidates of an anaphor in isolation, and thus cannot effectively capture the preference relationships between competing candidates for its learning and resolution. To deal with this problem, we propose a twin-candidate model for anaphora resolution. The main idea behind the model is to recast anaphora resolution as a preference classification problem. Specifically, the model learns a classifier that determines the preference between competing candidates, and, during resolution, chooses the antecedent of a given anaphor based on the ranking of the candidates. We present in detail the framework of the twin-candidate model for anaphora resolution. Further, we explore how to deploy the model in the more complicated coreference resolution task. We evaluate the twin-candidate model in different domains using the Automatic Content Extraction data sets. The experimental results indicate that our twin-candidate model is superior to the single-candidate model for the task of pronominal anaphora resolution. For the task of coreference resolution, it also performs equally well, or better.

1. Introduction

Anaphora is reference to an entity that has been previously introduced into the discourse (Jurafsky and Martin 2000). The referring expression used is called the **anaphor** and the expression being referred to is its **antecedent**. The anaphor is usually used to refer to the same entity as the antecedent; hence, they are **coreferential** with each other. The process of determining the antecedent of an anaphor is called **anaphora resolution**. As a key problem in discourse and language understanding, anaphora resolution is crucial in many natural language applications, such as machine translation, text summarization, question answering, information extraction, and so on. In recent

---

  [*] 21 Heng Mui Keng Terrace, Singapore, 119613. E-mail: xiaofengy@i2r.a-star.edu.sg.
[**] 21 Heng Mui Keng Terrace, Singapore, 119613. E-mail: sujian@i2r.a-star.edu.sg.
  [†] 3 Science Drive 2, Singapore 117543. E-mail: tancl@comp.nus.edu.sg.

years, supervised learning approaches have been widely applied to anaphora resolution, and they have achieved considerable success (Aone and Bennett 1995; McCarthy and Lehnert 1995; Connolly, Burger, and Day 1997; Kehler 1997; Ge, Hale, and Charniak 1998; Soon, Ng, and Lim 2001; Ng and Cardie 2002b; Strube and Mueller 2003; Luo et al. 2004; Ng et al. 2005).

The strength of learning-based anaphora resolution is that resolution regularities can be automatically learned from annotated data. Traditionally, learning-based approaches to anaphora resolution adopt the single-candidate model, in which the potential antecedents (i.e., **antecedent candidates**) are considered in isolation for both learning and resolution. In such a model, the purpose of classification is to determine if a candidate is the antecedent of a given anaphor. A training or testing instance is formed by an anaphor and each of its candidates, with features describing the properties of the anaphor and the individual candidate. During resolution, the antecedent of an anaphor is selected based on the classification results for each candidate.

One assumption behind the single-candidate model is that whether a candidate is the antecedent of an anaphor is completely independent of the other competing candidates. However, anaphora resolution can be more accurately represented as a ranking problem in which candidates are ordered based on their preference and the best one is the antecedent of the anaphor (Jurafsky and Martin 2000). The single-candidate model, which only considers the candidates of an anaphor in isolation, is incapable of effectively capturing the preference relationship between candidates for its training. Consequently, the learned classifier cannot produce reliable results for preference determination during resolution.

To deal with this problem, we propose a twin-candidate learning model for anaphora resolution. The main idea behind the model is to recast anaphora resolution as a preference classification problem. The purpose of the classification is to determine the preference between two competing candidates for the antecedent of a given anaphor. In the model, an instance is formed by an anaphor and two of its antecedent candidates, with features used to describe their properties and relationships. The antecedent is selected based on the judged preference among the candidates.

In the article we focus on two issues about the twin-candidate model. In the first part, we will introduce the framework of the twin-candidate model for anaphora resolution, including detailed training procedures and resolution schemes. In the second part, we will further explore how to deploy the twin-candidate model in the more complicated task of coreference resolution. We will present an empirical evaluation of the twin-candidate model in different domains, using the Automatic Content Extraction (ACE) data sets. The experimental results indicate that the twin-candidate model is superior to the single-candidate model for the task of pronominal anaphora resolution. For the coreference resolution task, it also performs equally well, or better.

2. Related Work

To our knowledge, the first work on the twin-candidate model for anaphora resolution was proposed by Connolly, Burger, and Day (1997). Their work relied on a set of features that included lexical type, grammatical role, recency, and number/gender/semantic agreement, and employed a simple linear search scheme to choose the most preferred candidate. Their system produced a relatively low accuracy rate for pronoun resolution (55.3%) and definite NP resolution (37.4%) on a set of selected news articles. Iida et al. (2003) used the twin-candidate model (called the *tournament model* in their work) to perform Japanese zero-anaphora resolution. They utilized the same linear

scheme to search for antecedents. Compared with Connolly, Burger, and Day (1997), they adopted richer features in which centering information was incorporated to capture contextual knowledge. Their system achieved an accuracy of around 70% on a data set drawn from a corpus of newspaper articles. Both of these studies were carried out on uncommon data sets, which makes it difficult to compare their results with other baseline systems. In contrast to the previous work, we will explore the twin-candidate model comprehensively by describing the model in more detail, trying more effective resolution schemes, deploying the model in the more complicated coreference resolution task, performing more extensive experiments, and evaluating the model in more depth.

Denis and Baldridge (2007) proposed a pronoun resolution system that directly used a ranking learning algorithm (based on Maximal Entropy) to train a preference classifier for antecedent selection. They reported an accuracy of around 72–76% for the different domains in the ACE data set. In our study, we will also investigate the solution of using a general ranking learner (e.g., Ranking-SVM). By comparison, the twin-candidate model is applicable to any discriminative learning algorithm, no matter whether it is capable of ranking learning or not. Moreover, as the model is trained and tested on pairwise candidates, it can effectively capture various relationships between candidates for better preference learning and determination.

Ng (2005) presented a ranking model for coreference resolution. The model focused on the preference between the potential partitions of NPs, instead of the potential antecedents of an NP as in our work. Given an input document, the model first employed n pre-selected coreference resolution systems to generate n candidate partitions of NPs. The model learned a preference classifier (trained using Ranking-SVM) that could distinguish good and bad partitions during testing. The best rank partition would be selected as the resolution output of the current text. The author evaluated the model on the ACE data set and reported an F-measure of 55–69% for the different domains. Although ranking-based, Ng's model is quite different from ours as it operates at the cluster-level whereas ours operates at the mention-level. In fact, the result of our twin-candidate system can be used as an input to his model.

3. The Twin-Candidate Model for Anaphora Resolution

3.1 The Single-Candidate Model

Learning-based anaphora resolution uses a machine learning method to obtain $p(ante(C_k)|ana, C_1, C_2, \ldots, C_n)$, the probability that a candidate C_k is the antecedent of the anaphor ana in the context of its antecedent candidates, $C_1, C_2, \ldots, C_n$. The single-candidate model assumes that the probability that C_k is the antecedent is only dependent on the anaphor ana and C_k, and independent of all the other candidates. That is:

$$p\left(ante(C_k) \mid ana, C_1, C_2, \ldots, C_n\right) = p\left(ante(C_k) \mid ana, C_k\right) \tag{1}$$

Thus, the probability of a candidate C_k being the antecedent can be approximated using the classification result on the instance describing the anaphor and C_k alone.

The single-candidate model is widely used in most anaphora resolution systems (Aone and Bennett 1995; Ge, Hale, and Charniak 1998; Preiss 2001; Strube and Mueller 2003; Kehler et al. 2004; Ng et al. 2005). In our study, we also build as the

Table 1
A sample text for anaphora resolution.

[$_1$ *Those figures*] are almost exactly what [$_2$ *the government*] proposed to [$_3$ *legislators*] in [$_4$ *September*]. If [$_5$ *the government*] can stick with [$_6$ *them*], [$_7$ *it*] will be able to halve this year's 120 billion ruble (US $193 billion) deficit.

Table 2
Training instances generated under the single-candidate model for anaphora resolution.

Anaphor	Training Instance	Label
[$_6$ *them*]	$i\{$[$_6$ *them*] , [$_1$ *Those figures*]$\}$	1
	$i\{$[$_6$ *them*] , [$_2$ *the government*]$\}$	0
	$i\{$[$_6$ *them*] , [$_3$ *legislators*]$\}$	0
	$i\{$[$_6$ *them*] , [$_4$ *September*]$\}$	0
	$i\{$[$_6$ *them*] , [$_5$ *the government*]$\}$	0
[$_7$ *it*]	$i\{$[$_7$ *it*] , [$_1$ *Those figures*]$\}$	0
	$i\{$[$_7$ *it*] , [$_3$ *legislators*]$\}$	0
	$i\{$[$_7$ *it*] , [$_4$ *September*]$\}$	0
	$i\{$[$_7$ *it*] , [$_5$ *the government*]$\}$	1
	$i\{$[$_7$ *it*] , [$_6$ *them*]$\}$	0

baseline a system for pronominal anaphora resolution based on the single-candidate model.

In the single-candidate model, an instance has the form of $i\{ana, cand_i\}$, where *ana* is an anaphor and *cand_i* is an antecedent candidate.[1] For training, instances are created for each anaphor occurring in an annotated text. Specifically, given an anaphor *ana* and its antecedent candidates, a set of negative instances (labeled "0") is formed by pairing *ana* and each of the candidates that is not coreferential with *ana*. In addition, a single positive instance (labeled "1") is formed by pairing *ana* and the closest antecedent, that is, the closest candidate that is coreferential with *ana*.[2] Note that it is possible that an anaphor has two or more antecedents, but we only create one positive instance for the closest antecedent as its reference relationship with the anaphor is usually the most direct and thus the most confident.

As an example, consider the text in Table 1.

Here, [$_6$ *them*] and [$_7$ *it*] are two anaphors. [$_1$ *Those figures*] and [$_5$ *the government*] are their closest antecedents, respectively. Supposing that the antecedent candidates of the two anaphors are just all their preceding NPs in the current text, the training instances to be created for the text segment are listed in Table 2.

---

1 In our study, we only consider anaphors whose antecedents are noun phrases. Typically, all the NPs preceding an anaphor can be taken as the initial antecedent candidates. For better learning and resolution, however, candidates can be filtered so that only those "confident" NPs, which occur in the specified search scope and meet constraints such as number/gender agreement, are considered. The details of candidate selection in our system will be discussed later in the section on experiments.

2 We assume that at least one antecedent exists in the candidate set of an anaphor. However, for real resolution, if none of the antecedents of an anaphor occur in the candidate set, we simply discard the anaphor and do not create any training instance for it.

Table 3
Feature set for pronominal anaphora resolution.

ana_Reflexive	whether the anaphor is a reflexive pronoun
ana_PronType	type of the anaphor if it is a pronoun (*he*, *she*, *it* or *they*?)
candi_Def	whether the candidate is a definite description
candi_Indef	whether the candidate is an indefinite NP
candi_Name	whether the candidate is a named entity
candi_Pron	whether the candidate is a pronoun
candi_FirstNP	whether the candidate is the first mentioned NP in the sentence
candi_Subject	whether the candidate is the subject of a sentence, the subject of a clause, or not.
candi_Oject	whether the candidate is the object of a verb, the object of a preposition, or not
candi_ParallelStruct	whether the candidate has an identical collocation pattern with the anaphor
candi_SentDist	the sentence distance between the candidate and the anaphor
candi_NearestNP	whether the candidate is the candidate closest to the anaphor in position

Note that for [$_7$ *it*], we do not use [$_2$ *the government*] to create a positive training instance as it is not the closest candidate that is coreferential with the anaphor.

A vector of features is specified for each training instance. The features may describe the characteristics of the anaphor and the candidate, as well as their relationships from lexical, syntactic, semantic, and positional aspects. Table 3 lists the features used in our study. All these features can be computed with high reliability, and have been proven effective for pronoun resolution in previous work.

Based on the generated feature vectors, a classifier is trained using a certain learning algorithm. During resolution, given a newly encountered anaphor, a test instance is formed for each of the antecedent candidates. The instance is passed to the classifier, which then returns a confidence value indicating the likelihood that the candidate is the antecedent of the anaphor. The candidate with the highest confidence is selected as the antecedent. For example, suppose [$_7$ *it*] is an anaphor to be resolved. Six test instances will be created for its six antecedent candidates, as listed in Table 4. The learned classifier is supposed to give the highest confidence to $i\{[_7\ it]\ ,\ [_5\ the\ government]\}$, indicating the candidate [$_5$ *the government*] is the antecedent of [$_7$ *it*].

3.2 A Problem with the Single-Candidate Model

As described, the assumption behind the single-candidate model is that the probability of a candidate being the antecedent of a given anaphor is completely independent of

Table 4
Test instances generated under the single-candidate model for anaphora resolution.

Anaphor	Test Instance
	$i\{[_7\ it]\ ,\ [_1\ Those\ figures]\}$
	$i\{[_7\ it]\ ,\ [_2\ the\ government]\}$
	$i\{[_7\ it]\ ,\ [_3\ legislators]\}$
[$_7$ *it*]	$i\{[_7\ it]\ ,\ [_4\ September]\}$
	$i\{[_7\ it]\ ,\ [_5\ the\ government]\}$
	$i\{[_7\ it]\ ,\ [_6\ them]\}$

the other competing candidates. However, for an anaphor, the determination of the antecedent is often subject to preference among the candidates (Jurafsky and Martin 2000). Whether a candidate is the antecedent depends on whether it is the "best" among the candidate set, that is, whether there exists no other candidate that is preferred over it. Hence, simply considering one candidate individually is an indirect and unreliable way to select the correct antecedent.

The idea of preference is common in linguistic theories on anaphora. Garnham (2001) summarizes different factors that influence the interpretation of anaphoric expressions. Some factors such as morphology (gender, number, animacy, and case) or syntax (e.g., the role of binding and commanding relations [Chomsky 1981]) are "eliminating," forbidding certain NPs from being antecedents. However, many others are "preferential," giving more preference to certain candidates over others; examples include:

- Sentence-based factors: Pronouns in one clause prefer to refer to the NP that is the subject of the previous clause (Crawley, Stevenson, and Kleinman 1990). Also, the NP that is the first-mentioned expression is preferred regardless of the syntactic and semantic role played by the referring expression (Gernsbacher and Hargreaves 1988).

- Stylistic factors: Pronouns preferentially take parallel antecedents that play the same role as the anaphor in their respective clauses (Grober, Beardsley, and Caramazza 1978; Stevenson, Nelson, and Stenning 1995).

- Discourse-based factors: Items currently in focus are the prime candidates for providing antecedents for anaphoric expressions. According to centering theory (Grosz, Joshi, and Weinstein 1995), each utterance has a set of forward-looking centers that have higher preference to be referred to in later utterances. The forward-looking centers can be ranked based on grammatical roles or other factors.

- Distance-based factors: Pronouns prefer candidates in the previous sentence compared with those two or more sentences back (Clark and Sengul 1979).

As a matter of fact, "eliminating" factors could also be considered "preferential" if we think of the act of eliminating candidates as giving them low preference.

Preference-based strategies are also widely seen in earlier manual approaches to pronominal anaphora resolution. For example, the SHRDLU system by Winograd (1972) prefers antecedent candidates in the subject position over those in the object position. The system by Wilks (1973) prefers candidates that satisfy selectional restrictions with the anaphor. Hobbs's algorithm (Hobbs 1978) prefers candidates that are closer to the anaphor in the syntax tree, and the RAP algorithm (Lappin and Leass 1994) prefers candidates that have a high salience value computed by aggregating the weights of different factors.

During resolution, the single-candidate model does select an antecedent based on preference by using classification confidence for candidates; that is, the higher confidence value the classifier returns, the more likely the candidate is preferred as the antecedent. Nevertheless, as the model considers only one candidate at a time during training, it cannot effectively capture the preference between candidates for classifier learning. For example, consider an anaphor and a candidate C_i. If there are no "better"

candidates in the candidate set, C_i is the antecedent and forms a positive instance. Otherwise, C_i is not selected as the antecedent and thus forms a negative instance. Simply looking at a candidate alone cannot explain this, and may possibly result in inconsistent training instances (i.e., the same feature vector but different class labels). Consequently, the confidence values returned by the learned classifier cannot reliably reflect the preference relationship between candidates.

3.3 The Twin-Candidate Model

To address the problem with the single-candidate model, we propose a twin-candidate model to handle anaphora resolution. As opposed to the single-candidate model, the model explicitly learns a preference classifier to determine the preference relationship between candidates. Formally, the model considers the probability that a candidate is the antecedent as the probability that the candidate is preferred over all the other competing candidates. That is:

$$p\left(ante(C_k) \mid ana, C_1, C_2, \ldots, C_n\right)$$
$$= p\left(C_k \succ \{C_1, \ldots, C_{k-1}, C_{k+1}, \ldots C_n\} \mid ana, C_1, C_2, \ldots, C_n\right) \tag{2}$$
$$= p(C_k \succ C_1, \ldots, C_k \succ C_{k-1}, C_k \succ C_{k+1}, \ldots, C_k \succ C_n \mid ana, C_1, C_2, \ldots, C_n)$$

Assuming that the preference between C_k and C_i is independent of the preference between C_k and the candidates other than C_i, we have:

$$p(C_k \succ C_1, \ldots, C_k \succ C_{k-1}, C_k \succ C_{k+1}, \ldots, C_k \succ C_n \mid ana, C_1, C_2, \ldots, C_n)$$
$$= \prod_{1 < i < n, i \neq k} p(C_k \succ C_i \mid ana, C_k, C_i) \tag{3}$$

Thus:

$$\ln p\left(ante(C_k) \mid ana, C_1, C_2, \ldots, C_n\right)$$
$$= \sum_{1 < i < n, i \neq k} \ln p(C_k \succ C_i \mid ana, C_k, C_i) \tag{4}$$

This suggests that the probability that a candidate C_k is the antecedent can be estimated using the classification results on the set of instances describing C_k and each of the other competing candidates. To do this, we learn a classifier that, given any two candidates of a given anaphor, can determine which one is preferred to be the antecedent of the anaphor. The final antecedent is identified based on the classified preference relationships among the candidates. This is the main idea of the twin-candidate model.

In such a model, each instance consists of three elements: $i\{ana, C_i, C_j\}$, where *ana* is an anaphor, and C_i and C_j are two of its antecedent candidates. The class label of an instance represents the preference between the two candidates for the antecedent, for example, "01" indicating C_j is preferred over C_i and "10" indicating C_i is preferred. Being trained with instances built based on this principle, the classifier is capable of determining the preference between any two candidates of a given anaphor by returning

Table 5
A sample text for anaphora resolution.

[1 *Those figures*] are almost exactly what [2 *the government*]
proposed to [3 *legislators*] in [4 *September*]. If [5 *the government*]
can stick with [6 *them*], [7 *it*] will be able to halve this year's
120 billion ruble (US $193 billion) deficit.

a class label, either "01" or "10", accordingly. In the next section, we will introduce in detail a system based on the twin-candidate model for anaphora resolution.

3.4 Framework of the Twin-Candidate Model

3.4.1 Instance Representation. In the twin-candidate model, an instance takes the form $i\{ana, C_i, C_j\}$, where *ana* is an anaphor and C_i and C_j are two of its antecedent candidates. We stipulate that C_j should be closer to *ana* than C_i in position (i.e., $i < j$). An instance is labeled "10" if C_i is preferred over C_j as the antecedent, or "01" if otherwise.

A feature vector is associated with an instance, and it describes different properties and relationships between *ana* and each of the candidates, C_i or C_j. In our study, the system with the twin-candidate model adopts the same feature set as the baseline system with the single-candidate model (shown in Table 3). The difference is that a feature for the single candidate, *candi_X*, has to be replaced by a pair of features for the twin candidates, *candi1_X* and *candi2_X*. For example, feature *candi_Pron*, which describes whether a candidate is a pronoun, will be replaced by two features *candi1_Pron* and *candi2_Pron*, which describe whether C_i and C_j are pronouns, respectively.

3.4.2 Training Instances Creation. To learn a preference classifier, a training instance for an anaphor should be composed of two candidates with an explicit preference relationship, for example, one being an antecedent and the other being a non-antecedent. A pair of candidates that are both antecedents or both non-antecedents are not suitable for instance creation because their preference cannot be explicitly represented for training, although it does exist.

Based on this idea, during training, for an encountered anaphor *ana*, we take the closest antecedent, C_{ante}, as the anchor candidate.[3] C_{ante} is paired with each of the candidates C_{nc} that is not coreferential with *ana*. If C_{ante} is closer to *ana* than C_{nc}, an instance $i\{ana, C_{nc}, C_{ante}\}$ is created and labeled "01". Otherwise, if C_{nc} is closer, an instance $i\{ana, C_{ante}, C_{nc}\}$ is created and labeled "10" instead.

Consider again the sample text given in Table 1, which is repeated in Table 5. For the anaphor [7 *it*], the closest antecedent, [5 *the government*] (denoted as NP_5), is chosen as the anchor candidate. It is paired with the four non-coreferential candidates (i.e., NP_1, NP_3, NP_4, and NP_6) to create four training instances. Among them, the instances formed with NP_1, NP_3 or NP_4 are labeled "01" and the one with NP_6 is labeled "10". Table 6 lists all the training instances to be generated for the text.

3.4.3 Classifier Generation. Based on the feature vectors for the generated training instances, a classifier can be trained using a discriminative learning algorithm. Given a test instance $i\{ana, C_i, C_j\}$ ($i < j$), the classifier is supposed to return a class label of "10",

---

3 If no antecedent is found in the candidate set, we do not generate any training instance for the anaphor.

Table 6
Training instances generated under the twin-candidate model for anaphora resolution.

Anaphor	Training Instance	Label
	$i\{[_6\ them], [_1\ Those\ figures], [_2\ the\ government]\}$	10
$[_6\ them]$	$i\{[_6\ them], [_1\ Those\ figures], [_3\ legislators]\}$	10
	$i\{[_6\ them], [_1\ Those\ figures], [_4\ September]\}$	10
	$i\{[_6\ them], [_1\ Those\ figures], [_5\ the\ government]\}$	10
	$i\{[_7\ it], [_1\ Those\ figures], [_5\ the\ government]\}$	01
	$i\{[_7\ it], [_3\ legislators], [_5\ the\ government]\}$	01
$[_7\ it]$	$i\{[_7\ it], [_4\ September], [_5\ the\ government]\}$	01
	$i\{[_7\ it], [_5\ the\ government], [_6\ them]\}$	10

indicating that C_i is preferred over C_j for the antecedent of *ana*, or "01", indicating that C_j is preferred.

3.4.4 Antecedent Identification. After training, the preference classifier can be used to resolve anaphors. The process of determining the antecedent of a given anaphor, called **antecedent identification**, could be thought of as a tournament, a competition in which many participants play against each other in individual matches. The candidates are like players in a tournament. A series of matches between candidates is held to determine the champion of the tournament, that is, the final antecedent of the anaphor under consideration. Here, the preference classifier is like the referee who judges which candidate wins or loses in a match.

If an anaphor has only one antecedent candidate, it is resolved to the candidate directly. For anaphors that have more than one candidate, two possible schemes can be employed to find the antecedent.

Tournament Elimination Tournament Elimination is a type of tournament where the loser in a match is immediately eliminated. Such a scheme is also applicable to antecedent identification. In the scheme, candidates are compared linearly from the beginning to the end. Specifically, the first candidate is compared with the second one, forming a test instance, which is then passed to the classifier to determine the preference. The "losing" candidate that is judged less preferred by the classifier is eliminated and never considered. The winner, that is, the preferred candidate, is compared with the third candidate. The process continues until all the candidates are compared, and the candidate that wins in the last comparison is selected as the antecedent.

For demonstration, we use the text in Table 5 as a test example. Suppose we have a "perfect" classifier that can correctly determine the preference between candidates. That is, the candidates that are coreferential with the anaphor will be classified as preferred over those that are not. (If the two candidates are both coreferential or both non-coreferential with the anaphor, the one closer to the anaphor in position is preferred.) To resolve the anaphor $[_7\ it]$, the candidate NP_1 is first compared with NP_2. The formed instance is classified as "01", indicating NP_2 is preferred. Thus, NP_1 is eliminated and NP_2 continues to compete with NP_3 and NP_4 until it fails in the comparison with NP_5. Finally, NP_5 beats NP_6 in the last match and is selected as the antecedent. All the test instances to be generated in sequence for the resolution of $[_6\ them]$ and $[_7\ it]$ are listed in Table 7.

The Tournament Elimination scheme has a computational complexity of $O(N)$, where N is the number of the candidates. Thus, it enables a relatively large number

Table 7
Test instances generated under the twin-candidate model with the Tournament Elimination scheme.

Anaphor	Test Instance	Result
	$i\{[_6 \text{ them}], [_1 \text{ Those figures}], [_2 \text{ the government}]\}$	10
$[_6 \text{ them}]$	$i\{[_6 \text{ them}], [_1 \text{ Those figures}], [_3 \text{ legislators}]\}$	10
	$i\{[_6 \text{ them}], [_1 \text{ Those figures}], [_4 \text{ September}]\}$	10
	$i\{[_6 \text{ them}], [_1 \text{ Those figures}], [_5 \text{ the government}]\}$	10
	$i\{[_7 \text{ it }], [_1 \text{ Those figures}], [_2 \text{ the government}]\}$	01
	$i\{[_7 \text{ it }], [_2 \text{ the government}], [_3 \text{ legislators}]\}$	10
$[_7 \text{ it }]$	$i\{[_7 \text{ it }], [_2 \text{ the government}], [_4 \text{ September}]\}$	10
	$i\{[_7 \text{ it }], [_2 \text{ the government}], [_5 \text{ the government}]\}$	01
	$i\{[_7 \text{ it }], [_5 \text{ the government}], [_6 \text{ them}]\}$	10

of candidates to be processed. However, as our twin-candidate model imposes no constraints that enforce transitivity of the preference relation, the preference classifier would likely output $C_1 \succ C_2$, $C_2 \succ C_3$, and $C_3 \succ C_1$. Hence, it is unreliable to eliminate a candidate once it happens to lose in one comparison, without considering all of its winning/losing results against the other candidates.

Round Robin In Section 3.3, we have shown that the probability that a candidate is the antecedent can be calculated using the preference classification results between the candidate and its opponents. The candidate with the highest preference is selected as the antecedent, that is:

$$Antecedent(ana) = \arg_i \max p\left(ante(C_i) \mid ana, C_1, C_2, \ldots, C_n\right)$$

$$\propto \arg_i \max \sum_{j \neq i} CF(i\{ana, C_i, C_j\}, C_i) \tag{5}$$

where $CF(i\{ana, C_i, C_j\}, C_i)$ is the confidence with which the classifier determines C_i to be preferred over C_j as the antecedent of *ana*. If we define the score of C_i as:

$$Score(C_i) = \sum_{j \neq i} CF(i\{ana, C_i, C_j\}, C_i) \tag{6}$$

Then, the most preferred candidate is the candidate that has the maximum score. If we simply use 1 to denote the result that C_i is classified as preferred over C_j, and -1 if C_j is preferred otherwise, then:

$$Score(C_i) = |\{C_j | C_i \succ C_j\}| - |\{C_j | C_j \succ C_i\}| \tag{7}$$

That is, the score of a candidate is the number of the opponents to which it is preferred, less the number of the opponents to which it is less preferred. To obtain the scores, the antecedent candidates are compared with each other. For each candidate, its comparison

Table 8
Test instances generated under the twin-candidate model with the Round Robin scheme.

Anaphor	Test Instance	Result
	$i\{[_7\ it], [_1\ \textit{Those figures}], [_2\ \textit{the government}]\}$	01
	$i\{[_7\ it], [_1\ \textit{Those figures}], [_3\ \textit{legislators}]\}$	01
	$i\{[_7\ it], [_1\ \textit{Those figures}], [_4\ \textit{September}]\}$	01
	$i\{[_7\ it], [_1\ \textit{Those figures}], [_5\ \textit{the government}]\}$	01
	$i\{[_7\ it], [_1\ \textit{Those figures}], [_6\ \textit{them}]\}$	01
	$i\{[_7\ it], [_2\ \textit{the government}], [_3\ \textit{legislators}]\}$	10
	$i\{[_7\ it], [_2\ \textit{the government}], [_4\ \textit{September}]\}$	10
$[_7\ it]$	$i\{[_7\ it], [_2\ \textit{the government}], [_5\ \textit{the government}]\}$	01
	$i\{[_7\ it], [_2\ \textit{the government}], [_6\ \textit{them}]\}$	10
	$i\{[_7\ it], [_3\ \textit{legislators}], [_4\ \textit{September}]\}$	01
	$i\{[_7\ it], [_3\ \textit{legislators}], [_5\ \textit{the government}]\}$	01
	$i\{[_7\ it], [_3\ \textit{legislators}], [_6\ \textit{them}]\}$	01
	$i\{[_7\ it], [_4\ \textit{September}], [_5\ \textit{the government}]\}$	01
	$i\{[_7\ it], [_4\ \textit{September}], [_6\ \textit{them}]\}$	01
	$i\{[_7\ it], [_5\ \textit{the government}], [_6\ \textit{them}]\}$	10

result against every other candidate is recorded. Its score increases by one if it wins a match, or decreases by one if it loses. The candidate with the highest score is selected as the antecedent.

Antecedent identification carried out in such a way corresponds to a type of tournament called Round Robin in which each participant plays every other participant once, and the final champion is selected based on the winning–losing records of the players. In contrast to the Elimination scheme, the Round Robin scheme is more reliable in that the preference of a candidate is determined by overall comparisons with the other competing candidates. The computational complexity of the scheme is $O(N^2)$, where N is the number of the candidates.

To illustrate this, consider the example in Table 5 again. The test instances to be generated for resolving the anaphor $[_7\ it]$ are listed in Table 8. As shown, each of the candidates is compared with every other competing candidate. The scores of the candidates are summarized in Table 9. Here, the candidate NP_5 beats all the opponents in the comparisons and obtains the maximum score of five. Thus it will be selected as the antecedent.

An extension of the above Round Robin scheme is called the Weighted Round Robin scheme. In the weighted version, the confidence values returned by the classifier,

Table 9
Scores for the candidates under the Round Robin scheme.

	NP_1	NP_2	NP_3	NP_4	NP_5	NP_6	Score
NP_1		−1	−1	−1	−1	−1	−5
NP_2	+1		+1	+1	−1	+1	+3
NP_3	+1	−1		−1	−1	−1	−3
NP_4	+1	−1	+1		−1	−1	−1
NP_5	+1	+1	+1	+1		+1	+5
NP_6	+1	−1	+1	+1	−1		+1

Table 10
Statistics for the training and testing data sets.

		NWire	NPaper	BNews
Train	# Tokens	85k	72k	67k
	# Files	130	76	216
Test	# Tokens	20k	18k	18k
	# Files	29	17	51

instead of the simple 0 and 1, are employed to calculate the score of a candidate based on the formula

$$Score(C_i) = \sum_{C_i \succ C_j} CF(C_i \succ C_j) - \sum_{C_k \succ C_i} CF(C_k \succ C_i) \qquad (8)$$

Here, CF is the confidence value that the classifier returns for the corresponding instance.

3.5 Evaluation

3.5.1 Experimental Setup. We used the ACE (Automatic Content Extraction)[4] coreference data set for evaluation. All the experiments were done on the ACE-2 V1.0 corpus. It contains two data sets, training and devtest, which were used for training and testing, respectively. Each of these sets is further divided into three domains: newswire (NWire), newspaper (NPaper), and broadcast news (BNews). Statistics for the data sets are summarized in Table 10.

For both training and resolution, a raw input document was processed by a pipeline of NLP modules including a Tokenizer, Part-of-Speech tagger, NP chunker, Named-Entity (NE) Recognizer, and so on. These preprocessing modules were meant to determine the boundary of each NP in a text, and to provide the necessary information about an NP for subsequent processing. Trained and tested on the UPEN WSJ TreeBank, the POS tagger (Zhou and Su 2000) could obtain an accuracy of 97% and the NP chunker (Zhou and Su 2000) could produce an F-measure above 94%. Evaluated for the MUC-6 and MUC-7 Named-Entity task, the NER module (Zhou and Su 2002) could provide an F-measure of 96.6% (MUC-6) and 94.1% (MUC-7).

In our experiments, we focused on the resolution of the third-personal pronominal anaphors, including *she*, *he*, *it*, *they* as well as their morphologic variants (such as *her*, *his*, *him*, *its*, *itself*, *them*, etc.). For both training and testing, we considered all the pronouns that had at least one preceding NP in their respective annotated coreferential chains. We used the accuracy rate as the evaluation metric, and defined it as follows:

$$Accuracy = \frac{\text{number of anaphors being correctly resolved}}{\text{total number of anaphors to be resolved}} \qquad (9)$$

Here, an anaphor is deemed "correctly resolved" if the found antecedent is in the co-referential chain of the anaphor.

---

4 See http://www.itl.nist.gov/iad/894.01/tests/ace for a detailed description of the ACE program.

Table 11
Statistics of the training instances generated for the pronominal anaphora resolution task.

		NWire	NPaper	BNews
Single-Candidate	0 instances	8,200	11,648	6,037
	1 instances	1,241	1,466	1,291
Twin-Candidate	01 instances	6,899	9,861	5,004
	10 instances	1,301	1,787	1,033

For pronoun resolution, the distance between the closest antecedent and the anaphor is usually short, predominantly (98% for the current data set) limited to only one or two sentences (McEnery, Tanaka, and Botley 1997). For this reason, given an anaphor, we only took the NPs occurring within the current and previous two sentences as initial antecedent candidates. The candidates with mismatched number and gender agreement were filtered automatically from the candidate set. Also, pronouns or NEs that disagreed in person with the anaphor were removed in advance. For training, there were 1,241 (NWire), 1,466 (NPaper), and 1,291 (BNews) anaphors found with at least one antecedent in the candidate set. For testing, the numbers were 313 (NWire), 399 (NPaper), and 271 (BNews). On average, an anaphor had nine antecedent candidates.

Table 11 summarizes the statistics of the training instances as well as the class distribution. Note that for the single-candidate model, the number of "1" instances was identical to the number of anaphors in the training data, because we only used the closest antecedents of anaphors to create the positive instances. The number of "0" instances was equal to the total number of "01" and "10" training instances for the twin-candidate model.

We examined three different learning algorithms: C5 (Quinlan 1993), Maximum Entropy (Berger, Della Pietra, and Della Pietra 1996), and SVM (linear kernel) (Vapnik 1995),[5] using the software See5,[6] OpenNlp.MaxEnt,[7] and SVM-light,[8] respectively. All the classifiers were learned with the default learning parameters set in the respective learning software.

3.5.2 Results and Discussions. Table 12 lists the performance of the different anaphora resolution systems with the single-candidate (SC) and the twin-candidate (TC) models. For the TC model, two antecedent identification schemes, Tournament Elimination and Round Robin, were compared.

From the table, we can see that our baseline system with the single-candidate model can obtain accuracy of up to 72.9% (NWire), 77.1% (NPaper), and 74.9% (BNews).

---

5 As MaxEnt learns a probability model, we used the returned probability as the confidence of a candidate being the antecedent. For C5, the confidence value of a candidate was estimated based on the following smoothed ratio:

$$CF = \frac{p+1}{t+2}$$

where c was the number of positive instances and t was the total number of instances stored in the corresponding leaf node. For SVM, the returned value was used as the confidence value: the lower (maybe negative) the less confident.

6 http://www.rulequest.com/see5-info.html
7 http://MaxEnt.sourceforge.net/
8 http://svmlight.joachims.org/

Table 12
Accuracy in percent for the pronominal anaphora resolution.

		NWire	NPaper	BNews	Average
C5	SC	71.6	75.6	69.5	72.7
	TC				
	- Elimination	71.6	81.3	74.5	76.4
	- Round Robin	72.9	81.3	74.9	**76.9**
	- Weighted Round Robin	72.9	80.5	75.6	76.7
MaxEnt	SC	72.9	77.1	74.9	75.2
	TC				
	- Elimination	75.1	79.1	77.5	**77.4**
	- Round Robin	75.1	79.1	77.5	**77.4**
	- Weighted Round Robin	75.7	78.6	77.1	77.3
SVM	SC	72.9	77.3	74.2	75.1
	TC				
	- Elimination	73.5	82.0	78.9	78.5
	- Round Robin	74.4	82.0	78.9	**78.7**
	- Weighted Round Robin	74.6	79.3	78.2	77.5
	Rank_SVM	73.5	79.3	76.4	76.7

The average accuracy is comparable to that reported by Kehler et al. (2004) (around 75%), who also used the single-candidate model to do pronoun resolution with similar features (using MaxEnt) on the ACE data sets. By contrast, the systems with the twin-candidate model are able to achieve accuracy of up to 75.7% (NWire), 82.0% (NPaper), and 78.9% (BNews). The average accuracy is 76.9% for C5, 77.4% for MaxEnt, and 78.7% for SVM, which is statistically significantly[9] better than the results of the baselines (4.2%, 2.2%, and 3.6% in accuracy). These results confirm our claim that the twin-candidate model is more effective than the single-candidate model for the task of pronominal anaphora resolution.

We see no significant difference between the accuracy rates (less than 1.0% accuracy) produced by the two antecedent identification schemes, Tournament Elimination and Round Robin. This is in contrast to our belief that the Round Robin scheme, which is more reliable than the Tournament Elimination, should lead to much better results. One possible reason could be that the classifier in our systems can make a correct preference judgement (with accuracy above 92% as in our test) in the cases where one candidate is the antecedent and the other is not. As a consequence, the simple linear search can find the final antecedent as well as the Round Robin method. These results suggest that we can use the Elimination scheme in a practical system to make antecedent identification more efficient. (Recall that the Elimination scheme requires complexity of $O(N)$, instead of $O(N^2)$ as in Round Robin.)

Ranking-SVM In our experiments, we were particularly interested in comparing the results using the twin-candidate model and those directly using a preference learning algorithm. For this purpose, we built a system based on Ranking-SVM (Joachims 2002), an extension of SVM capable of preference learning.

---

9 Throughout our experiments, the significance was examined by using the paired *t*-test, with $p < 0.05$.

The system uses a similar framework to the single-candidate-based system. For training, given an anaphor, a set of instances is created for each of the antecedent candidates. To learn the preference between competing candidates, a "query-ID" is specified for each training instance in such a way that the instances formed by the candidates of the same anaphor bear the same query-ID. The label of an instance represents the rank of the candidate in the candidate set; here, "1" for the instances formed by the candidates that are the antecedents, and "0" for the instances formed by the others. The training instances are associated with features as defined in Table 3, to which the Ranking-SVM algorithm is then applied to generate a preference classifier. During resolution, for each candidate of a given anaphor, a test instance is formed and passed to the learned classifier, which in turn returns a value to represent the rank of the candidate among all the candidates. The anaphor is resolved to the one with the highest value.

In fact, if we look into the learning mechanism of Ranking-SVM, we can find that the algorithm will, in the background, pair any two instances that have the same query-ID but different rank labels. This is quite similar to the twin-candidate model, which creates an instance by putting together two candidates with different preferences. However, one advantage of the twin-candidate model is that it can explicitly record various relationships between two competing candidates, for example, "which one of the two candidates is closer to the anaphor in position/syntax/semantics?"[10] Such inter-candidate information can make the preference between candidates clearer, and thus facilitate both preference learning and determination. In contrast, Ranking-SVM, which constructs instances in the single-candidate form, cannot effectively capture this kind of information.

The last line of Table 12 shows the results from such a system based on Ranking-SVM. We can see that the system achieves an average accuracy of 76.7%, statistically significantly better than the baseline system with the single-candidate model by 1.6% (0.4% for NWire, 2.0% for NPaper, and 2.2% for BNews). The results lend support to our claim that the preference relationships between candidates, if taken into consideration for classifier training, can lead to better resolution performance. Still, we observe that our twin-candidate model beats Ranking-SVM in average accuracy by 1.8% (Elimination scheme) and 2.0% (Round Robin).

Decision Tree One advantage of the C5 learning algorithm is that the generated classifier can be easily interpreted by humans, and the importance of the features can be visually illustrated. In Figures 1 and 2, we show the decision trees (top four levels) output by C5 for the NWire domain, based on the single-candidate and the twin-candidate models, respectively. As the twin-candidate model uses a larger pool of features, the tree for the twin-candidate model is more complicated (180 nodes) than the one for the single-candidate model (36 nodes).

From the two trees, we can see that both models rely on similar features such as lexical, positional, and grammatical properties for pronoun resolution. However, we can see that the preferential factors (e.g., subject preference, parallelism preference, and distance preference as discussed in Section 3.2) are more clearly presented in the twin-candidate-based tree. For example, if two candidates are both pronouns, the twin-candidate-based tree will suggest that the one closer to the anaphor has a higher preference to be the antecedent. By contrast, such a preference relationship has to be implicitly represented

---

10 In the current work, we only consider the positional relationship between candidates by stipulating that $i < j$ for an instance $i\{ana, C_i, C_j\}$. In our future work, we will explore more inter-candidate relationships that are helpful for preference determination.

```
candi_Pron = 1:
:...candi_SentDist = 0: 1 (329/42)
:    candi_SentDist = 2: 0 (207/34)
:    candi_SentDist = 1:
:    :...ana_Type = Pron_SHE: 1 (22/4)
:        ana_Type = Pron_HE: 1 (166/45)
:        ana_Type = Pron_IT: 0 (46/8)
:        ana_Type = Pron_THEY:
:        :...candi_NearestNP = 0: 0 (39/11)
:            candi_NearestNP = 1: 1 (14/2)
candi_Pron = 0:
:...candi_ParallelStruct = 1: 1 (14/2)
    candi_ParallelStruct = 0:
    :...candi_NearestNP = 1:
        :...candi_Subject = NO: 0 (369/71)
        :    candi_Subject = SUBJ_MAIN: 1 (106/19)
        :    candi_Subject = SUBJ_CLAUSE: 1 (82/24)
        candi_NearestNP = 0:
        :...candi_Name = 0: 0 (6617/256)
            candi_Name = 1: ...
            :...candi_SentDist = 1: 0 (553/69)
                candi_SentDist = 2: 0 (491/8)
                candi_SentDist = 0:
                :...candi_Object = OBJ_VERB: 0 (48/22)
                    candi_Object = OBJ_PREP: 0 (82/16)
                    candi_Object = NO: ...
```

Figure 1
Decision tree generated for pronoun resolution under the single-candidate model. For feature
ana_Type, the values PRON_SHE,PRON_SHE,PRON_SHE, and PRON_THEY represent whether
the anaphor is a pronoun such as *she*, *he*, *it*, and *they*, respectively. For *candi_Subject*, the values
SUBJ_MAIN, SUBJ_CLAUSE and NO represent whether the candidate is the subject of a main
sentence, or the subject of a clause, or not. For *candi_Object*, the values OBJ_VERB, OBJ_PREP, and
NO represent whether the candidate is the object of a verb, a preposition, or not, respectively.
For other features, 0 and 1 represent yes/no.

in the single-candidate-based tree, with different confidence values being assigned to
the candidates in different sentences.

Learning Curve In our experiments, we were also concerned about how training
data size might influence anaphora resolution performance. For this purpose, we di-
vided the anaphors in the training documents into 10 batches, and then performed
resolution using the classifiers trained with 1, 2, ..., 10 batches of anaphors. Figure 3
plots the learning curves of the systems with the single-candidate model and the twin-
candidate model (Round Robin scheme) for the NPaper domain. Each accuracy rate
shown in the figure is the average of the results from three trials trained on different
anaphors.

From the figure we can see that both the single-candidate model and the twin-
candidate model reach their peak performance with around six batches (around
880 anaphors). As shown, the twin-candidate model is not apparently superior to
the single-candidate model when the size of the training data is small (below two
batches, 290 anaphors). This is due to the fact that the number of features in the twin-
candidate model is nearly double that in the single-candidate model. As a result, the
twin-candidate model requires more training data than the single-candidate model to
avoid the data sparseness problem. Nevertheless, it does not need too much training
data to beat the latter; it can produce the accuracy rates consistently higher than the

```
candi1_Pron = 1:
:...candi2_Pron = 1: 1 (106/9)
:    candi2_Pron = 0:
:    :...candi2_Subject = SUBJ_MAIN:
:        :...candi2_SentDist = 1: 10 (17/1)
:        :    candi2_SentDist = 2: 1 (4/2)
:        :    candi2_SentDist = 0: ...
:        candi2_Subject = NO:
:        :...candi2_Name = 0: ...
:        :    candi2_Name = 1: ...
:        candi2_Subject = SUBJ_CLAUSE:
:        :...candi1_Object = OBJ_VERB: 1 (14/2)
:            candi1_Object = OBJ_PREP: 10 (3)
:            candi1_Object = NO: ...
candi1_Pron = 0:
:...candi1_ParallelStruct = 1: 10 (32/2)
    candi1_ParallelStruct = 0:
    :...candi2_Object = NO: 1 (5411/154)
        candi2_Object = OBJ_PREP:
        :...candi1_Subject = SUBJ_CLAUSE: ...
        :    candi1_Subject = NO: ...
        :    candi1_Subject = SUBJ_MAIN: ...
        candi2_Object = OBJ_PREP:
        :...candi1_Subject = SUBJ_CLAUSE: ...
            candi1_Subject = NO: ...
            candi1_Subject = SUBJ_MAIN: ...
```

Figure 2
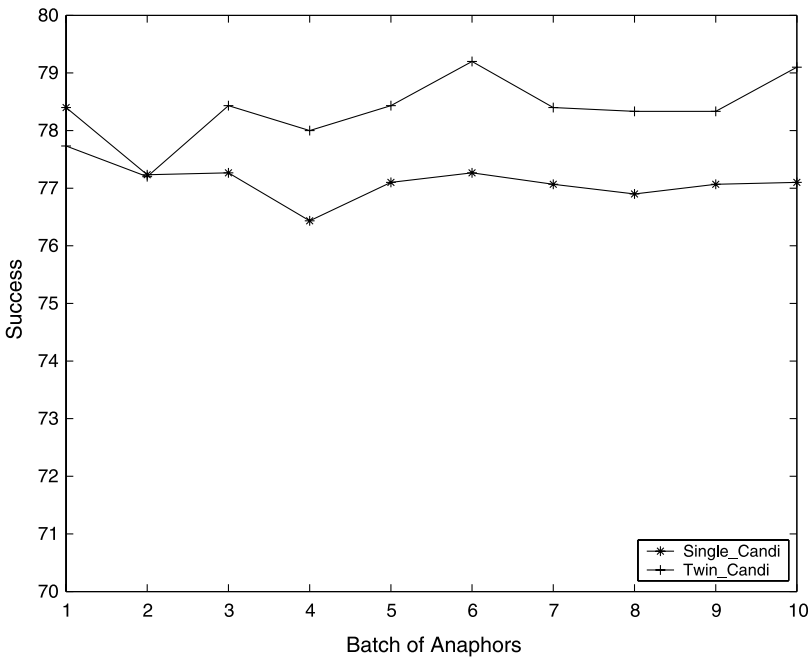Decision tree generated for pronoun resolution under the twin-candidate model.

Figure 3
Learning curves of different models for pronominal anaphora resolution in the NPaper Domain
(120 anaphors per batch).

Table 13
A sample text for coreference resolution.

[$_1$ *Globalstar*] still needs to raise [$_2$ $600 million], and
[$_3$ *Schwartz*] said [$_4$ *that company*] would try to raise [$_5$ *the money*] in [$_6$ *the debt market*].

single-candidate model when trained with more than two batches of anaphors. This figure further demonstrates that the twin-candidate model is reliable and effective for the pronominal anaphora resolution task.

4. Deploying the Twin-Candidate Model to Coreference Resolution

One task that is closely related to anaphora resolution is **coreference resolution**, the process of identifying all the coreferential expressions in texts.[11] Coreference resolution is different from anaphor resolution. The latter focuses on how an anaphor can be successfully resolved, and the resolution is done on given anaphors. The former, in contrast, focuses on how the NPs that are coreferential with each other can be found correctly and completely, and the resolution is done on all possible NPs. In a text, many NPs, especially the non-pronouns, are non-anaphors that have no antecedent to be found in the previous text. Hence, the task of coreference resolution is a more complicated challenge than anaphora resolution, as a solution should not only be able to resolve an anaphor to the correct antecedent, but should also refrain from resolving a non-anaphor. In this section, we will explore how to deploy the learning models for anaphor resolution in the coreference resolution task. As pronouns are usually anaphors, we will focus mainly on the resolution of non-pronouns.

4.1 Coreference Resolution Based on the Single-Candidate Model

In practice, the single-candidate model can be applied to coreference resolution directly, using the similar training and testing procedures to those used in anaphora resolution (described in Section 2).

For training, we create "0" and "1" training instances for each encountered anaphor, that is, the NP that is coreferential with at least one preceding NP. Specifically, given an anaphor and its antecedent candidates, a positive instance is generated for the closest antecedent and a set of negative instances is generated for each of the candidates that is not coreferential with the anaphor.[12]

Consider the text in Table 13 as an example. In the text, [$_4$ *that company*] and [$_5$ *the money*] are two anaphors, with [$_1$ *Globalstar*] and [$_2$ *$600 million*] being their antecedents, respectively. Table 14 lists the training instances to be created for this text.

---

11 In our study, we only consider within-document noun phrase coreference resolution.
12 In some coreference resolution systems (Soon, Ng, and Lim 2001; Ng and Cardie 2002b), only the non-coreferential candidates occurring between the closest antecedent and the anaphor are used to create negative instances. In the experiments, we found that these sampling strategies for negative instances led to a trade-off between recall and precision, but no significant difference in the overall F-measure.

Table 14
Training instances generated under the single-candidate model for coreference resolution.

Anaphor	Training Instance	Label
[4 *that company*]	i{[4 *that company*] , [1 *Globalstar*]}	1
	i{[4 *that company*] , [2 *$600 million*]}	0
	i{[4 *that company*] , [3 *Schwartz*]}	0
[5 *the money*]	i{[5 *the money*] , [1 *Globalstar*]}	0
	i{[5 *the money*] , [2 *$600 million*]}	1
	i{[5 *the money*] , [3 *Schwartz*]}	0
	i{[5 *the money*] , [4 *that company*]}	0

Table 15
Feature set for coreference resolution.

ana_Def	whether the possible anaphor is a definite description
ana_Indef	whether the possible anaphor is an indefinite NP
ana_Name	whether the possible anaphor is a named entity
candi_Def	whether the candidate is a definite description
candi_Indef	whether the candidate is an indefinite description
candi_Name	whether the candidate is a named-entity
candi_SentDist	the sentence distance between the possible anaphor and the candidate
candi_NameAlias	whether the candidate and the candidate are aliases for each other
candi_Appositive	whether the possible anaphor and the candidate are in an appositive structure
candi_NumberAgree	whether the possible anaphor and the candidate agree in number
candi_GenderAgree	whether the possible anaphor and the candidate agree in gender
candi_HeadStrMatch	whether the possible anaphor and the candidate have the same head string
candi_FullStrMatch	whether the possible anaphor and the candidate contain the same strings (excluding the determiners)
candi_SemAgree	whether the possible anaphor and the candidate belong to the same semantic category in WordNet

In Table 15, we list the features used in our study for coreference resolution, which are similar to those proposed in Soon, Ng, and Lim's (2001) system.[13] All these features are domain independent and the values can be computed with low cost but high reliability.

After training, the learned classifier can be directly used for coreference resolution. Given an NP to be resolved, a test instance is generated for each of its antecedent candidates. The classifier, being given the instance, will determine the likelihood that the candidate is the antecedent of the possible anaphor. If the confidence is below a pre-specified threshold, the candidate is discarded. In the case where none of the candidates have a confidence higher than the threshold, the current NP is deemed a

---

13 As we focus on coreference resolution for non-pronouns, we do not use the feature that describes whether or not the NP to be resolved is a pronoun. Also, we do not use the feature that describes whether or not a candidate is a pronoun, because, as will be discussed together with the experiments, a pronoun is not taken as an antecedent candidate for a non-pronoun to be resolved.

non-anaphor and left unresolved. Otherwise, it is resolved to the candidate with the highest confidence.[14]

4.2 Coreference Resolution Based on the Twin-Candidate Model

The twin-candidate model presented in the previous section focuses on the preference between candidates. The model will always select a "best" candidate as the antecedent, even if the current NP is a non-anaphor. To deal with this problem, we will teach the preference classifier how to identify non-anaphors, by incorporating non-anaphors to create a special class of training instances. For resolution, if the newly learned classifier returns the special class label, we will know that the current NP is a non-anaphor, and no preference relationship holds between the two candidates under consideration. In this way, the twin-candidate model is capable of carrying out both antecedent identification and anaphoricity determination by itself, and thus can be deployed for coreference resolution directly. In this section, we will describe the modified training and resolution procedures of the twin-candidate model.

4.2.1 Training. As with anaphora resolution, an instance of the twin-candidate model for coreference resolution takes the form $i\{ana, C_i, C_j\}$, where *ana* is a possible anaphor, and C_j and C_j are two of its antecedent candidates $(i < j)$. The feature set is similar to that for the single-candidate model as defined in Table 15, except that a *candi_X* feature is replaced by a pair of features, *cand1_x* and *candi2_x*, for the two competing candidates, respectively.

During training, if an encountered NP is an anaphor, we create "01" or "10" training instances in the same way as in the original learning framework. If the NP is a non-anaphor, we do the following:

- From the antecedent candidates,[15] randomly select one as the anchor candidate.

- Create a set of instances by pairing the anchor candidate and each of the other non-coreferential candidates.

The instances formed by the non-anaphors are labeled "00."

Consider the sample text in Table 13. For the two anaphors [4 *that company*] and [5 *the money*], we create the "01" and "10" instances as usual. For the non-anaphors [3 *Schwartz*] and [6 *the debt market*], we generate two sets of "00" instances. Table 16 lists all the training instances for the text (supposing [1 *Globalstar*] and [2 *$600 million*] are the anchor candidates for [3 *Schwartz*] and [6 *the debt market*], respectively).

The "00" training instances are used together with the "01" and "10" ones to train a classifier. Given a test instance $i\{ana, C_i, C_j\}$ $(i < j)$, the newly learned classifier is supposed to return "01" (or "10"), indicating *ana* is an anaphor and C_i (or C_j) is preferred as its antecedent, or return "00", indicating *ana* is a non-anaphor and no preference exists between C_i and C_j.

---

14 Other clustering strategies are also available, for example, "closest-first" where a possible anaphor is resolved to the closest candidate with the confidence above the specified threshold, if any (Soon, Ng, and Lim 2001).

15 For a non-anaphor, we also take the preceding NPs as its antecedent candidates. We will discuss this issue later together with the experimental setup.

Table 16
Training instances generated under the twin-candidate model for coreference resolution.

Possible Anaphor	Training Instance	Label
[$_4$ *that company*]	i{[$_4$ *that company*], [$_1$ *Globalstar*], [$_2$ *\$600 million*]} i{[$_4$ *that company*], [$_1$ *Globalstar*], [$_3$ *Schwartz*]}	10 10
[$_5$ *the money*]	i{[$_5$ *the money*], [$_1$ *Globalstar*], [$_2$ *\$600 million*]} i{[$_5$ *the money*], [$_2$ *\$600 million*], [$_3$ *Schwartz*]} i{[$_5$ *the money*], [$_2$ *\$600 million*], [$_4$ *that company*]}	01 10 10
[$_3$ *Schwartz*]	i{[$_3$ *Schwartz*], [$_1$ *Globalstar*], [$_2$ *\$600 million*]}	00
[$_6$ *the debt market*]	i{[$_6$ *the debt market*], [$_1$ *Globalstar*], [$_2$ *\$600 million*]} i{[$_6$ *the debt market*], [$_2$ *\$600 million*], [$_3$ *Schwartz*]} i{[$_6$ *the debt market*], [$_2$ *\$600 million*], [$_4$ *that company*]} i{[$_6$ *the debt market*], [$_2$ *\$600 million*], [$_5$ *the money*]}	00 00 00 00

4.2.2 Antecedent Identification. Accordingly, we make a modification to the original Tournament Elimination and the Round Robin schemes:

Tournament Elimination Scheme As with anaphora resolution, given an NP to be resolved, candidates are compared linearly from the beginning to the end. If an instance for two competing candidates is classified as "01" or "10", the preferred candidate will be compared with subsequent competitors while the loser is eliminated immediately. If the instance is classified as "00", both the two candidates are discarded and the comparison restarts with the next two candidates.[16] The process continues until all the candidates have been compared. If both of the candidates in the last match are judged to be "00", the current NP is left unresolved. Otherwise, the NP will be resolved to the final winner, on the condition that the highest confidence that the winner has ever obtained is above a pre-specified threshold.

Round Robin Scheme In the Round Robin scheme, each candidate is compared with every other candidate. If two candidates are labeled "00" in a match, both candidates receive a penalty of -1 in their respective scores. If no candidate has a positive final score, then the NP is considered non-anaphoric and left unresolved. Otherwise, it is resolved to the candidate with the highest score as usual. Here, we can also use a threshold. That is, we will update the scores of the two candidates in a match if and only if the preference confidence returned by the classifier is higher than a pre-specified threshold.

In rare cases where an NP to be resolved has only one antecedent candidate, a pseudo-instance is created by pairing the candidate with itself. The NP will be resolved to the candidate unless the instance is labeled "00".

4.3 Evaluation

4.3.1 Experimental Setup. We used the same ACE data sets for coreference resolution evaluation, as described in the previous section for anaphora resolution. A raw input document was processed in advance by the same pipeline of NLP modules including

---

16 If only one candidate remains, it will be compared with the candidate eliminated last.

Table 17
Statistics of the training instances generated for coreference resolution (non-pronoun).

		NWire	NPaper	BNews
Single-Candidate	0 instances	78,191	105,152	33,748
	1 instances	3,197	3,792	2,094
Twin-Candidate	00 instances	296,000	331,957	159,752
	01 instances	50,499	70,433	21,170
	10 instances	27,692	34,719	12,578

POS-tagger, NP chunker, NE recognizer, and so on, to obtain all possible NPs and related information (see Section 3.5.1).

For evaluation, we adopted Vilain et al.'s (1995) scoring algorithm in which *recall* and *precision*[17] were computed by comparing the key chains (i.e., the annotated "standard" coreferential chains) and the response chains (i.e., the chains generated by the coreference resolution system).

As already mentioned, the twin-candidate model described in this section is mainly meant for non-pronouns that are often not anaphoric. To better examine the utility of the model in our experiments, we first focused on coreference resolution for non-pronominal NPs. The recall and precision to be reported were computed based on the response chains and the key chains from which all the pronouns are removed. We will later show the results of overall coreference resolution for whole NPs by combining the resolution of pronouns and non-pronouns.

In non-pronoun resolution, an anaphor and its antecedent do not often occur a short distance apart as they do in pronoun resolution. For this reason, during training, we took as antecedent candidates all the preceding non-pronominal NPs[18] in the current and previous four sentences; while during testing, we used all the preceding non-pronouns, regardless of distance, as candidates.[19] The statistics of the training instances for each data set are summarized in Table 17.

Again, we examined the three learning algorithms: C5, MaxEnt, and SVM.[20] As both the single-candidate and the twin-candidate models used a threshold to block low-confidence coreferential pairs, we performed three-fold cross-evaluation on the training data to determine the thresholds for the coreference resolution systems.

4.3.2 Results and Discussions. Table 18 lists the results for the different systems on the non-pronominal NP coreference resolution. We used as the baseline the system with the single-candidate model described in Section 4.1. As mentioned, the system was trained

---

17 The overall F-measure was defined as

$$\frac{2 * Recall * Precision}{Recall + Precision}$$

18 As suggested in Ng and Cardie (2002b), we did not include pronouns in the candidate set of a non-pronoun, because a pronoun is usually anaphoric and cannot give much information about the entity to which it refers.

19 Unlike in the case of pronoun resolution, we did not filter candidates that had mismatched number/gender agreement as these constraints are not reliable for non-pronoun resolution (e.g., in our data set, around 15% of coreferential pairs do not agree in number). Instead, we took these factors as features (see Table 15) and let the learning algorithm make the preference decision.

20 For SVM, we employed the *one-against-all* aggregation method for the 3-class learning and testing.

Table 18
Recall (R), Precision (P), and F-measure (F) in percent for coreference resolution (non-pronoun).

		NWire			NPaper			BNews		
		R	P	F	R	P	F	R	P	F
C5	SC									
	- baseline	63.3	48.1	54.7	63.8	42.2	50.8	63.5	53.7	**58.2**
	- with non-anaphors	40.9	81.5	54.4	39.8	81.4	53.4	35.1	76.8	48.2
	TC									
	- Elimination	50.8	63.0	56.2	56.6	60.1	58.3	44.6	71.2	54.9
	- Round Robin	58.7	57.9	**58.3**	56.5	60.5	**58.4**	49.0	70.1	57.7
MaxEnt	SC									
	- baseline	62.1	52.3	56.8	56.4	58.8	57.6	61.8	54.1	57.7
	- with non-anaphors	59.6	54.0	56.7	54.2	62.6	58.1	53.8	58.4	56.0
	TC									
	- Elimination	59.1	55.4	57.2	52.2	69.0	**59.5**	53.5	61.9	57.4
	- Round Robin	58.7	55.9	**57.2**	53.4	65.9	59.0	54.3	62.8	**58.3**
SVM	SC									
	- baseline	64.1	49.0	55.5	65.5	42.1	51.3	63.5	53.7	58.2
	- with non-anaphors	42.3	70.0	52.7	40.0	76.6	52.5	35.7	77.0	48.8
	TC									
	- Elimination	57.8	53.2	55.4	51.7	56.5	54.0	63.3	53.8	58.2
	- Round Robin	54.3	56.9	**55.6**	56.1	58.1	**57.1**	63.7	53.8	**58.3**

on the instances formed by anaphors. For better comparison with the twin-candidate model, we built another single-candidate-based system in which the non-anaphors were also incorporated for training. Specifically, for each encountered non-anaphor during training, we created a set of "0" instances by pairing the non-anaphor with each of the candidates. These instances were added to the original instances formed by anaphors to learn a classifier,[21] which was then applied for the resolution as usual.

The results for the two single-candidate based systems are listed in Table 18. When trained with the instances formed only by anaphors, the system could achieve recall above 60% and precision of around 50% for the three domains. When trained with the instances formed by both anaphors and non-anaphors, the system yielded a significant improvement in precision. In the case of using C5 and SVM, the system is capable of producing precision rates of up to 80%. The increase in precision is reasonable since the classifier tends to be stricter in blocking non-anaphors. Unfortunately, however, at the same time recall drops significantly, and no apparent improvement can be observed in the resulting overall F-measure.

When trained with non-anaphors incorporated, the systems with the twin-candidate model, described in Section 4.2, are capable of yielding higher precision against the baseline. Although recall also drops at the same time, the increase in precision can compensate it well: We observe that in most cases, the system with the twin-candidate model can achieve a better F-measure than the baseline system with the single-candidate model. Also, the improvement is statistically significant (t-test, $p < 0.05$) in the NWire domain when C5 is used (3.6%), and in the NPaper domain

---

21 The statistics of the "0" instances shown in Table 17 become 392,646, 455,167, and 207,667 for NWire, NPaper, and BNews, respectively.

when any of the three learning algorithms, C5 (5.0%), MaxEnt (1.4%), and SVM (4.6%), is used. These results suggest that our twin-candidate model can effectively identify non-anaphors and block their invalid resolution, without affecting the accuracy of determining antecedents for anaphors.

Compared with the pronoun resolution described in the previous section, here we find that for non-pronoun resolution the superiority of the twin-candidate model against the single-candidate model is not apparent. In some domains such as BNews, the difference between the two models is not statistically significant. One possible explanation is that for non-pronoun resolution, the features that really matter are quite limited, that is, NameAlias, String-Matching, and Appositive (we will later show this in the decision trees). A candidate that has any one of these features is most likely the antecedent, regardless of the other competing candidates. In this situation, the single-candidate model, which considers candidates in isolation, does as well as the twin-candidate model. Still, the results suggest that the twin-candidate model is suitable for both resolution tasks, no matter whether the features involved are strongly indicative (as with non-pronoun resolution) or not (as with pronoun resolution).

As with anaphora resolution, we do not observe any apparent performance difference between the two twin-candidate identification schemes, Tournament Elimination and Round Robin. The Round Robin scheme performs better than Elimination when trained using C5 and SVM, by up to 2.8% and 2.9% in F-measure, respectively. However, the Elimination scheme, when trained using MaxEnt, is capable of performing equally well or slightly better (0.5% F-measure) than the Round Robin scheme.

Recall vs. Precision As discussed, the results in Table 18 show different recall and precision patterns for different systems. The baseline system with the single-candidate model tends to yield higher recall while the system with the twin-candidate model tends to produce higher precision. Thus, a fairer comparison of the two systems is to examine the precision rates that these systems achieve under the same recall rates. For this purpose, in Figure 4, we plot the variant recall and precision rates that the two systems are capable of obtaining (tested using MaxEnt, Round Robin scheme, for the NPaper domain), focusing on precision rates above 50% and recall rates above 40%. From the figure, we find that the system with the twin-candidate model achieves higher precision for recall rates ranging from 40% and 55%, and performs equally well for recall rates above 55%, which further proves the reliability of our twin-candidate model for coreference resolution.

Decision Trees In Figures 5 and 6, we show the two decision trees (NWire domain) generated by the systems with the single-candidate model and the twin-candidate model, respectively. The tree from the single-candidate model contains only 13 nodes, considerably smaller than that from the twin-candidate model, which contains around 1.2k nodes. From the figure, we can see that both models heavily rely on string-matching, name-alias, and appositive features to perform non-pronoun resolution, in contrast to pronoun resolution where lexical and positional features seem more important (as shown in Figures 1 and 2).

Learning Curves In our experiments, we were also interested in evaluating the resolution performance of the two learning models on different quantities of training data. Figure 7 plots the learning curves for the systems using the single-candidate model and the system using the twin-candidate model (NPaper domain). The F-measure is averaged over three random trials trained on 5, 10, 15, ... documents. Consistent with the curves for the anaphora resolution task as depicted in Figure 3, the system with the twin-candidate model outperforms the one with the single-candidate model on a small amount of training data (less than five documents). When more data is available,
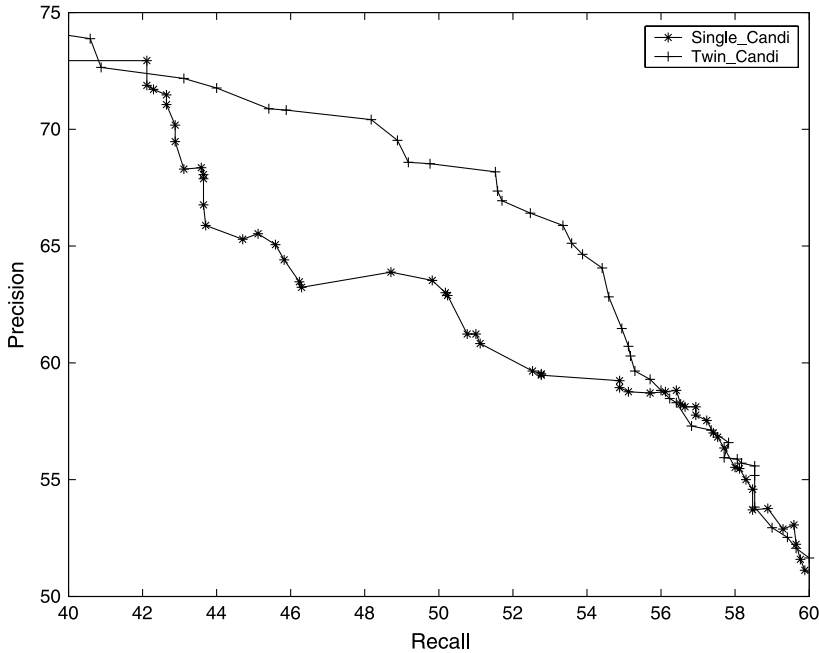
Figure 4
Various recall (%) and precision (%) of different models for non-pronoun resolution.

```
candi_HeadMatch = 0:
:...candi_Appositive = 1: 1 (75/1)
:   candi_Appositive = 0:
:   :...candi_NameAlias = 0: 0 (79264/1389)
:       candi_NameAlias = 1: 1 (88/29)
candi_HeadMatch = 1:
:...candi_NameAlias = 1: 1 (1198/84)
    candi_NameAlias = 0:
    :...candi_Name = 1: 0 (122/55)
        candi_Name = 0:
        :...ana_Name = 0: 1 (595/104)
            ana_Name = 1: 0 (46/15)
```

Figure 5
Decision tree generated for non-pronoun resolution under the single-candidate model.

the twin-candidate model also yields a consistently better F-measure than the single-candidate model.

Overall Coreference Resolution Having demonstrated the performance of the twin-candidate model on coreference resolution for non-pronouns, we now further examine overall coreference resolution for whole NPs, combining both pronoun resolution and non-pronoun resolution. Specifically, given an input test document, we check each encountered NP from beginning to end. If it is a pronoun,[22] we use

---

22 We identify the pleonastic use of *it* in advance (79.2% accuracy) using a set of predefined pattern rules based on regular expressions. The first-person and second-person pronouns are heuristically resolved to the closest pronoun of the same type or a speaker nearby, if any, with an average 61.8% recall and 79.5% precision.

```
cand1_HeadMatch = 1:
:...candi2_Appositive = 1: 1 (9/1)
:    candi2_Appositive = 0:
:    :...candi2_NameAlias = 1:
:        :...candi2_Name = 0: 0 (4)
:        :   candi2_Name = 1: 1 (83/25)
:        candi2_NameAlias = 0:
:        :...candi2_HeadMatch = 1: ...
:            candi2_HeadMatch = 0: ...
candi1_HeadMatch = 0:
:...candi2_HeadMatch = 1:
    :...candi2_Name = 1: 1 (17238/212)
    :   candi2_Name = 0:
    :   :...candi2_NameAlias = 1: 0 (131/18)
    :        candi2_NameAlias = 0: ...
    candi2_HeadMatch = 0:
    :...candi2_Appositive = 1: 1 (1809/3)
        candi2_Appositive = 0:
        :...candi1_NameAlias = 1: ...
            candi1_NameAlias = 0: ...
```

Figure 6
Decision tree generated for non-pronoun resolution under the twin-candidate model.
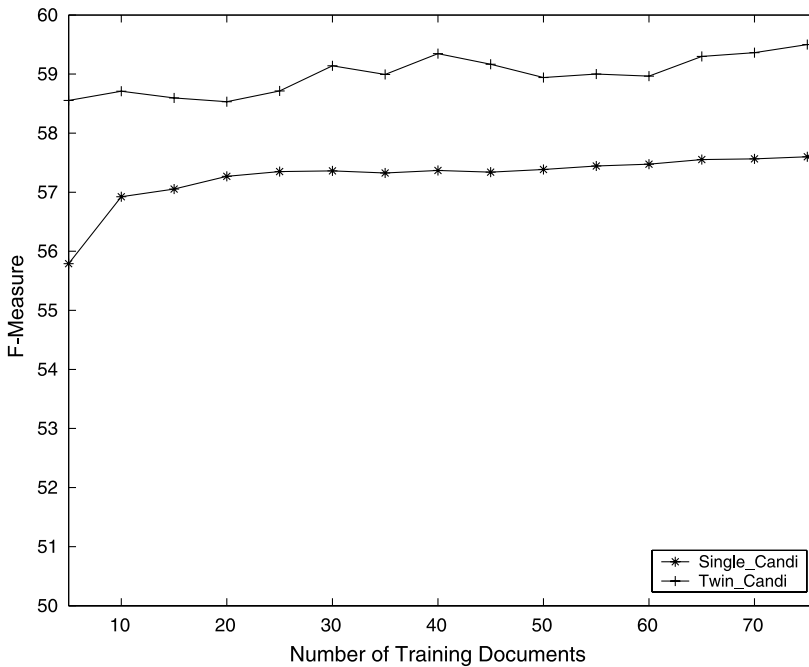

Figure 7
Learning curves of different models for non-pronoun resolution.

the pronominal anaphora resolution systems, as described in the previous section, to resolve it to an antecedent. Otherwise, we use the non-pronoun coreference resolution systems described in this section to resolve the NP to an antecedent, if any is found. All the coreferential pairs are put together in a coreferential chain. The recall and precision rates are computed by comparing the standard key chains and generated response chains using Vilain et al.'s (1995) algorithm.

Table 19
Recall (R), Precision (P), and F-Measure (F) in percent for coreference resolution.

		NWire			NPaper			BNews		
		R	P	F	R	P	F	R	P	F
C5	SC	62.2	52.6	57.0	64.9	50.6	56.9	62.9	58.5	60.6
	TC									
	- Elimination	53.8	65.9	59.2	61.2	64.4	62.8	53.1	70.9	60.7
	- Round Robin	59.0	61.2	**60.1**	62.0	64.3	**63.1**	56.0	69.9	**62.2**
MaxEnt	SC	60.7	56.0	58.3	60.8	62.2	61.5	63.8	60.6	62.2
	TC									
	- Elimination	59.5	59.2	**59.3**	58.6	67.8	62.9	59.3	66.4	62.7
	- Round Robin	60.6	57.9	59.2	59.4	69.2	**63.3**	61.7	64.5	**63.0**
SVM	SC	62.3	53.3	57.5	66.2	50.5	57.3	64.7	60.1	62.3
	TC									
	- Elimination	57.6	57.0	57.3	58.5	62.6	60.5	65.0	60.6	62.7
	- Round Robin	56.0	60.6	**58.2**	60.4	63.6	**62.0**	65.4	60.7	**63.0**

Table 19 lists the coreference resolution results of the systems with different learning models. We observe that the results for overall coreference resolution are better than those of non-pronoun coreference resolution as shown in Table 18, which is due to the comparatively high accuracy of the resolution of pronouns.

In line with the previous results for pronoun resolution and non-pronoun resolution, the twin-candidate model outperforms the single-candidate model in coreference resolution for whole NPs. Consider the system trained with MaxEnt as an example. The single-candidate-based system obtains F-measures of 58.3%, 61.5%, and 62.2% for the NWire, NPaper, and BNews domains.[23] By comparison, the twin-candidate-based system (Round Robin scheme) can achieve F-measures of 59.2%, 63.3%, and 63.0% for the three domains. The improvement over the single-candidate model in F-measure (0.9%, 1.8%, and 0.8%) is larger than that for non-pronoun resolution (0.4%, 1.4%, and 0.6% as shown in Table 18), owing to the higher gains obtained from pronoun resolution. For the systems trained using C5 and SVM, similar patterns of performance improvement may be observed.

5. Conclusion

In this article, we have presented a twin-candidate model for learning-based anaphora resolution. The traditional single-candidate model considers candidates in isolation, and thus cannot accurately capture the preference relationships between competing candidates to provide reliable resolution. To deal with this problem, our proposed twin-candidate model recasts anaphora resolution as a preference classification problem. It learns a classifier that can explicitly determine the preference between competing candidates, and then during resolution, choose the antecedent of an anaphor based on the ranking of the candidates.

---

23 The results are comparable to the baseline system by Ng (2005), which also uses the single-candidate model and is capable of F-measures of 50.1%, 62.1%, and 57.5% for the three domains, respectively.

We have introduced in detail the framework of the twin-candidate model for anaphora resolution, including instance representation, training procedure, and the antecedent identification scheme. The efficacy of the twin-candidate model for pronominal anaphora resolution has been evaluated in different domains, using ACE data sets. The experimental results show that the model yields statistically significantly higher accuracy rates than the traditional single-candidate model (up to 4.2% in average accuracy rate), suggesting that the twin-candidate model is superior to the latter for pronominal anaphora resolution.

We have further investigated the deployment of the twin-candidate model in the more complicated coreference resolution task, where not all the encountered NPs are anaphoric. We have modified the model to make it directly applicable for coreference resolution. The experimental results for non-pronoun resolution indicate that the twin-candidate-based system performs equally well, and, in some domains, statistically significantly better than the single-candidate based systems. When combined with the results for pronoun resolution, the twin-candidate based system achieves further improvement against the single-candidate-based systems in all the domains.

A number of further contributions can be made by extending this work in new directions. Currently, we only adopt simple domain-independent features for learning. Our recent work (Yang, Su, and Tan 2005) suggests that more complicated features, such as statistics-based semantic compatibility, can be effectively incorporated in the twin-candidate model for pronoun resolution. In future work, we intend to provide a more in-depth investigation into the various kinds of knowledge that are suitable for the twin-candidate model. Furthermore, in our current work for coreference resolution, all the NPs preceding an anaphor are used as antecedent candidates, and all encountered non-anaphors in texts are incorporated without filtering into training instance creation. For more balanced training data and better classifier learning, we intend to explore some instance-sampling techniques, such as those proposed by Ng and Cardie (2002a), to remove in advance low-confidence candidates and the less informative non-anaphors. We hope that these efforts can further improve the performance of the twin-candidate model in both anaphora resolution and coreference resolution.

References
Aone, Ghinatsu and Scott W. Bennett. 1995. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 122–129, Cambridge, Massachusetts.

Berger, Adam L., Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Foris, Dordrecht, The Netherlands.

Clark, Herber H. and C. J. Sengul. 1979. In search of referents for noun phrases and pronouns. *Memory and Cognition*, 7:35–41.

Connolly, Dennis, John D. Burger, and David S. Day, 1997. A machine learning approach to anaphoric reference. In *New Methods in Language Processing*, pages 133–144, Taylor and Francis, Bristol, Pennsylvania.

Crawley, Rosalind A., Rosemary J. Stevenson, and David Kleinman. 1990. The use of heuristic strategies in the interpretation of pronouns. *Journal of Psycholinguistic Research*, 19:245–264.

Denis, Pascal and Jason Baldridge. 2007. A ranking approach to pronoun resolution. In *Proceedings of the 20th International Joint*

Conference on Artificial Intelligence (IJCAI), pages 1588–1593, Hyderabad, India.

Garnham, Alan, 2001. *Mental Models and the Interpretation of Anaphora*. Psychology Press Ltd., Hove, East Sussex, UK.

Ge, Niyu, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *Proceedings of the 6th Workshop on Very Large Corpora*, pages 161–171, Montreal, Quebec, Canada.

Gernsbacher, Morton A. and David Hargreaves. 1988. Accessing sentence participants: The advantage of first mention. *Journal of Memory and Language*, 27:699–717.

Grober, Ellen H., William Beardsley, and Alfonso Caramazza. 1978. Parallel function in pronoun assignment. *Cognition*, 6:117–133.

Grosz, Barbara J., Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Hobbs, Jerry. 1978. Resolving pronoun references. *Lingua*, 44:339–352.

Iida, Ryu, Kentaro Inui, Hiroya Takamura, and Yuji Matsumoto. 2003. Incorporating contextual cues in trainable models for coreference resolution. In *Proceedings of the 10th Conference of EACL, Workshop "The Computational Treatment of Anaphora,"* pages 23–30, Budapest, Hungary.

Joachims, Thorsten. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, Edmonton, Alberta, Canada.

Jurafsky, Daniel and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, New Jersey.

Kehler, Andrew. 1997. Probabilistic coreference in information extraction. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 163–173, Providence, Rhode Island.

Kehler, Andrew, Douglas Appelt, Lara Taylor, and Aleksandr Simma. 2004. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics annual meeting (NAACL)*, pages 289–296, Boston, MA.

Lappin, Shalom and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):525–561.

Luo, Xiaoqiang, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 135–142, Barcelona, Spain.

McCarthy, Joseph F. and Wendy G. Lehnert. 1995. Using decision trees for coreference resolution. In *Proceedings of the 14th International Conference on Artificial Intelligences (IJCAI)*, pages 1050–1055, Montreal, Quebec, Canada.

McEnery, A., I. Tanaka, and S. Botley. 1997. Corpus annotation and reference resolution. In *Proceedings of the ACL Workshop on Operational Factors in Practical Robust Anaphora Resolution for Unrestricted Texts*, pages 67–74, Madrid, Spain.

Ng, Hwee Tou, Yu Zhou, Robert Dale, and Mary Gardiner. 2005. Machine learning approach to identification and resolution of one-anaphora. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1105–1110, Edinburgh, Scotland.

Ng, Vincent. 2005. Machine learning for coreference resolution: From local classification to global ranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 157–164, Ann Arbor, Michigan.

Ng, Vincent and Claire Cardie. 2002a. Combining sample selection and error-driven pruning for machine learning of coreference rules. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 55–62, Philadelphia, PA.

Ng, Vincent and Claire Cardie. 2002b. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 104–111, Philadelphia, PA.

Preiss, Judita. 2001. Machine learning for anaphora resolution. Technical Report CS-01-10, University of Sheffield, Sheffield, England.

Quinlan, J. Ross. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Francisco, CA.

Soon, Wee Meng, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Stevenson, Rosemary J., Alexander W. R. Nelson, and Keith Stenning. 1995. The role of parallelism in strategies of pronoun comprehension. *Language and Speech*, 29:393–418.

Strube, Michael and Christoph Mueller. 2003. A machine learning approach to pronoun resolution in spoken dialogue. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 168–175, Sapporo, Japan.

Vapnik, Vladimir N. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY.

Vilain, Marc, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 45–52, San Francisco, CA.

Wilks, Yorick. 1973. *Preference Semantics*. Stanford AI Laboratory Memo AIM-206. Stanford University.

Winograd, Terry. 1972. *Understanding Natural Language*. Academic Press, New York.

Yang, Xiaofeng, Jian Su, and Chew Lim Tan. 2005. Improving pronoun resolution using statistics-based semantic compatibility information. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 165–172, Ann Arbor, MI.

Zhou, Guodong and Jian Su. 2000. Error-driven HMM-based chunk tagger with context-dependent lexicon. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 71–79, Hong Kong.

Zhou, Guodong and Jian Su. 2002. Named Entity recognition using a HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 473–480, Philadelphia, PA.