A TWIN-CANDIDATE MODEL FOR LEARNING BASED COREFERENCE RESOLUTION

YANG, XIAOFENG

NATIONAL UNIVERSITY OF SINGAPORE

2005

A TWIN-CANDIDATE MODEL FOR LEARNING BASED COREFERENCE RESOLUTION

YANG, XIAOFENG
(B.Eng. M.Eng., Xiamen University)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2005

Acknowledgments

First, I would like to take this opportunity to thank all the people who helped me to complete this thesis.

I would first like to thank my supervisor, Dr. Jian Su, for her guidance, knowledge, and invaluable supports all the way. I owe much to my co-supervisor, Dr. Chew Lim Tan, who gave me much good advice on my research and in particular, managed to provide his critical and careful proof-reading which significantly improved the presentation of this thesis. I am also grateful to my senior colleague, Dr. Guodong Zhou. I have benifitted a lot from his thoughtful comments and suggestions. And his NLP systems proved essential for my research work.

I would also like all my labmates at the Institute for Infocomm Research: Jinxiu Chen, Huaqing Hong, Dan Shen, Zhengyu Niu, Juan Xiao, Jie Zhang and many other people for making the lab a pleasant place to work, and making my life in Singapore a wonderful memeory.

Finally, I would like to thank my parents and my wife, Jinrong Zhuo, who provide the love and support I can always count on. They know my gratitude.

Contents

Summary

Coreference resolution is the process of finding multiple expressions which are used to refer to the same entity. In recent years, supervised machine learning approaches have been applied to this problem and achieved considerable success. Most of these approaches adopt the single-candidate model, that is, only one antecedent candidate is considered at a time when resolving a possible anaphor. The assumption behind the single-candidate model is that the reference relation between the anaphor and one candidate is independent of the other candidates. However, for coreference resolution, the selection of the antecedent is determined by the preference between the competing candidates. The single-candidate model, which only considers one candidate for its learning, cannot accurately represent the preference relationship between competing candidates.

With the aim to overcome the limitations of the single-candidate model, this thesis proposes an alternative twin-candidate model to do coreference resolution. The main idea behind the model is to recast antecedent selection as a preference classification problem. Specifically, the model will learn a classifier that can determine the preference between two competing candidates of a given anaphor, and then choose the antecedent based on the ranking of the candidates.

The thesis focuses on three issues related to the twin-candidate model.

First, it explores how to use the twin-candidate model to identify the antecedent from the set of candidates of an anaphor. In detail, it introduces the construction of the basic twin-candidate model including the instance representation, the training data creation and the classifier generation. Also, it presents and discusses several strategies for the antecedent selection.

Second, it investigates how to deploy the twin-candidate model to coreference resolution in which the anaphoricity of an encountered expression is unknown. It presents several possible solutions to make the twin-candidate applicable to coreference resolution. Then it proposes a modified twin-candidate model, which can do both antecedent selection and anaphoricity determination by itself and thus can be directly employed to do coreference resolution.

Third, it discusses how to represent the knowledge for preference determination in the twin-candidate model. It presents the organization of different types of knowledge, and then gives a detailed description of the definition and computation of the features used in the study.

The thesis evaluates the twin-candidate model on the newswire domain, using the MUC data set. The experimental results indicate that the twin-candidate model achieves better results than the single-candidate model in finding correct antecedents for given anaphors. Moreover, the results show that for coreference resolution, the modified twin-candidate model outperforms the single-candidate model as well as the basic twin-candidate model. The results also suggest that the preference knowledge used in the study is reliable for both anaphora resolution and coreference resolution.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

To make computers understand human languages is a key step to a successful intelligent system. Although it may sound easy for human beings, the task, known as Natural Language Processing (NLP), is still a very difficult challenge for computers. A system capable of processing natural languages should not only be able to analyze words, phrases and sentences, but also be able to correctly understand the structure and cohesion within the current dialogue or discourse. To achieve this more advanced goal, the system should have the capability to identify the coreference relations between different expressions in discourse.

Coreference accounts for cohesion in texts. Coreference resolution is the process of identifying, within or across documents, multiple expressions that are used to refer to the same entity in the world. As a key problem to discourse and language understanding, coreference resolution is crucial in many NLP applications, such as machine translation (MT), text summarization (TS), information extraction (IE), question answering (QA) and so on.

Coreference resolution has long been recognized as an important and difficult prob-

lem by researchers in linguistics, philosophy, psychology and computer science. The history of the study on coreference resolution could be dated back to 1960s−1970s (Bobrow, 1964; Charniak, 1972; Winograd, 1972; Woods et al., 1972). Much of the early work on coreference resolution heavily relies on syntax (Winograd, 1972; Hobbs, 1976; Hobbs, 1978; Sidner, 1979; Carter, 1987), semantics (Charniak, 1972; Wilks, 1973; Wilks, 1975; Carter, 1987; Carbonell and Brown, 1988), or discourse knowledge (Kantor, 1977; Lockman, 1978; Webber, 1978; Grosz, 1977; Sidner, 1978; Brennan et al., 1987). However, such knowledge is usually difficult to represent and process, and the encoding of the knowledge would require a large amount of human effort.

The need for a robust and inexpensive solution to build a practical NLP system encouraged researchers to turn to knowledge-poor approaches (Lappin and Leass, 1994; Kennedy and Boguraev, 1996; Williams et al., 1996; Baldwin, 1997; Mitkov, 1998). With the availability of corpora as well as sophisticated NLP tools, recent years have seen the application of statistical and AI techniques, especially machine learning techniques, in coreference resolution (Dagan and Itai, 1990; Aone and Bennett, 1995; McCarthy and Lehnert, 1995; Connolly et al., 1997; Kehler, 1997b; Ge et al., 1998; Cardie and Wagstaff, 1999; Soon et al., 2001; Ng and Cardie, 2002b). Among them, supervised learning approaches, in which the coreference resolution regularities could be automatically learned from annotated data, receive more and more research attention (Aone and Bennett, 1995; McCarthy and Lehnert, 1995; Connolly et al., 1997; Kehler, 1997b; Ge et al., 1998; Soon et al., 2001; Ng and Cardie, 2002b; Strube and Mueller, 2003; Luo et al., 2004; Yang et al., 2004a; Ng et al., 2005).

As with other learning based applications, before applying a specific learning algorithm to coreference resolution, we shall first design the learning model of the problem. For example, if we decide to recast coreference resolution as a classification problem, we have to consider how to represent the training and the testing instances, how to define the features for the instances, and how to use the learned classifier to

do the resolution.

Traditionally, the learning-based approaches to coreference resolution adopt the single-candidate model, in which the resolution task is recast as a binary classification problem. In the model, an instance is formed by an anaphor and one of its antecedent candidates. Features are used to describe the properties of the anaphor and the single candidate, as well as their relationships. The classification is to determine whether or not a candidate is coreferential to the anaphor in question. During resolution, the antecedent of a given anaphor is selected based on the classification result for each candidate, with a certain clustering strategy like *best-first* (Aone and Bennett, 1995; Ng and Cardie, 2002b; Yang et al., 2004a) or *closest-first* (Soon et al., 2001).

Nevertheless, the single-candidate model has problems in the following aspects:

First and foremost, **representation**. The single-candidate model represents coreference resolution as a simple "COREF-OR-NONCOREF" problem, assuming that the coreference relationship between an anaphor and one antecedent candidate is completely independent of the other competing candidates. However, the antecedent selection process could 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. The single-candidate model, which only considers one candidate of an anaphor at time, is incapable of capturing the preference relationship between the candidates.

Also, **resolution**. In the single-candidate model, the coreference between an anaphor and an antecedent candidate is determined independently without considering other candidates. Therefore, it would be possible that two or more candidates are judged as coreferential to the anaphor. How to select the antecedent from these "positive" candidates becomes a problem, as simply linking the anaphor to all these candidates significantly degrades the precision and the overall performance (Soon et al., 2001). The commonly used strategies to find the best candidate, such as *best-first*

and *closest-first*, are done in an ad-hoc manner and may not be the optimal from an empirical point of view (Ng, 2005).

1.2 Goals

To overcome the limitations of the single-candidate model, this thesis proposes a twin-candidate model to do coreference resolution.The main idea behind the twin-candidate model is to recast antecedent selection as a preference classification problem. That is, the classification is done between two competing candidates to determine their preference as the antecedent of a given anaphor, instead of being done on one individual candidate to determine its reference with the 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 final antecedent is selected based on the preference among the candidates.

The thesis will focus on three issues about the twin-candidate model:

How does the twin-candidate model work for antecedent selection?

As described, in the twin-candidate model, the purpose of classification is to determine the preference between two candidates. Now the issue is: How to train such a preference classifier? And how to use the classifier to select the antecedent? The thesis will describe in detail the basic construction of the twin-candidate model for antecedent selection, including the representation of the instances, the creation of the training data, the generation of the preference classifier, and the selection of the antecedent. Particularly, the thesis gives much emphasis on the antecedent selection strategies. It presents and compares different selection schemes including *elimination* and *round-robin*. The effectiveness of the twin-candidate model in antecedent selec-

tion for anaphors will be examined in the experiments.

How to deploy the twin-candidate model to coreference resolution?

The basic twin-candidate model focuses on selecting the most preferred candidate as the antecedent for a given anaphor. However, the model itself can not identify the anaphoricity of the expression to be resolved. That is, in coreference resolution the model always picks out a "best" candidate even though the encountered expression is a non-anaphor that has no antecedent in the candidate set. In order to make the twin-candidate model applicable to coreference resolution, the thesis presents several possible strategies, like using an additional anaphoricity determination module, using a candidate filter, and using a threshold. Then it proposes a modified twin-candidate model that uses a classifier learned on the training instances with non-anaphors being incorporated. The modified model is capable of doing non-anaphoricity determination and antecedent selection at the same time, and thus can be directly deployed to coreference resolution. The efficacy of the modified twin-candidate model for coreference resolution and its advantages over the other strategies will be analyzed in the experiments.

How to represent the knowledge for preference determination in the twin-candidate model?

In machine leaning approaches, knowledge is generally encoded in terms of features. The twin-candidate model organizes the features for preference determination in two ways. First, it puts together the two sets of features that respectively describe one of the two competing candidates under consideration, assuming the classifier could compare the features related to the two candidates and then make a preference deci-

sion. Second, the model uses a set of features to describe the relationships between the competing candidates. These inter-candidate features are capable of directly representing the preference factors between the candidates. With these features, the preference between two competing candidates becomes clearer for both learning and testing. In the thesis, a detailed description of the features adopted in our study will be given, and their utility for antecedent selection and coreference resolution will be evaluated in the experiments.

1.3 Overview of the Thesis

Chapter 2 gives the basic concepts related to coreference. It analyzes the properties of coreference and summarizes some common coreference phenomena occurring in natural language texts. Also, it describes the task of coreference resolution as well as evaluation methods commonly used for this task.

Chapter 3 surveys the previous research work on coreference resolution. The first part of the literature review focuses on the non-learning based work, including the knowledge-rich based approaches and more recent knowledge-poor based approaches. The second part concentrates on the machine learning based work, including those unsupervised-learning, supervised-learning and weakly-supervised-learning approaches. Advantages and disadvantages of these approaches are discussed in the chapter.

Chapter 4 discusses the possible learning models of coreference resolution. It begins by the comparison of the *all-candidate* model and the commonly adopted *single-candidate* model and shows the superiority of the latter over the former. Then it points out the problems of the single-candidate model in both representation and resolution, and then proposes the alternative *twin-candidate* model. It shows the rationale of the twin-candidate model and its advantages over the *single-candidate*

model.

Chapter 5 starts with the detailed description of the twin-candidate model and shows how it works for antecedent selection. It introduces the instance representation, training, and antecedent selection problems of the model. Then in the second part, it discusses how to deploy the twin-candidate model to do coreference resolution. Four feasible strategies are proposed to make the twin-candidate applicable to coreference resolution. Both pros and cons of these strategies are discussed.

Chapter 6 focuses on the knowledge representation issue of the twin-candidate model. The chapter first introduces the organization of the feature set, and then gives a detailed description of the features adopted in our study, including their definition and computation. Particularly, it emphasizes the inter-candidate features that are related to the relationships between candidates.

Chapter 7 presents the evaluation of the twin-candidate model. After introducing the coreference resolution system that is to be run in the experiments, the chapter first demonstrates the efficacy of the twin-candidate model in antecedent identification for anaphors. Then it shows the capability of the twin-candidate model in coreference resolution. In-depth analysis and discussion of the experimental results are given in the chapter.

Finally, Chapter 8 presents conclusions and suggests future work.

Chapter 2

Coreference and Coreference Resolution

Coreference resolution is the process of linking, within or across documents, multiple expressions which refer to the same entity in the world. It is a key problem to discourse and language understanding, and is crucial in many natural language applications, such as machine translation (MT), text summarization (TS), information extraction (IE), question answering (QA) and so on.

This chapter will present the background knowledge about coreference and the coreference resolution task. The first part of the chapter gives the basic notations and concepts of coreference, and summarizes some common coreference phenomena in discourse. The second part describes the task of coreference resolution and introduces the commonly adopted evaluation methods for this task.

2.1 Coreference

2.1.1 What is coreference?

What is coreference? Various definitions have been put forward in literature. From the perspective of computational linguistics, ***coreference*** *is the act of referring to the same referent in the real word.* (Mitkov, 2002). Two referring expressions that are used to refer to the same entity are said to ***co-refer*** or to be ***coreferential*** (Jurafsky and Martin, 2000).

Referring expressions could be noun phrases or verb phrases, occurring within a document or across different documents. In our thesis, we will only focus on the within-document noun phrase (NP) coreference.

Put in a computational way. Suppose we define NP(n) if n is an NP expression, ENTITY(e) if e is an entity, and REF(n, e) if n is referred to e. Then coreference COREF is a relation such that

$$\begin{aligned} \forall\ n1 \forall\ n2, \quad & \text{NP}(n1),\ \text{NP}(n2),\ \text{COREF}(n1,\ n2) \\ \Leftrightarrow\ & \exists e,\ \text{ENTITY}(e),\ \text{REF}(n1,\ e),\ \text{REF}(n2,\ e) \end{aligned} \tag{2.1}$$

For better understanding, consider the following text,

(**Eg 2.1**) [$_1$ Microsoft Corp.] announced [$_3$ [$_2$ its] new CEO] [$_4$ yesterday]. [$_5$ The company] said [$_6$ he] will ...

There are six expressions in the above text segment. Among them, the first expression [$_1$ Microsoft Corp.] refers to an entity which is a company and has the name "Microsoft". From the context, the pronoun [$_2$ its] and the definite noun phrase [$_5$ The company] both refer to the same entity, i.e. the company of Microsoft.

Therefore, the three expressions [$_1$ Microsoft Corp.], [$_2$ its] and [$_5$ The company] have coreference relations with one another. Similarly, the noun phrase [$_3$ its new CEO] and the pronoun [$_6$ he] both refer to the certain human being who is the CEO newly appointed by Microsoft, and thus are coreferential to each other. In contrast, there is no expression that refers to the time that is referred to by [$_4$ yesterday], so there exists no coreference relation between [$_4$ yesterday] and any other expression in the text.

2.1.2 Coreference: An Equivalence Relation

Coreference is an equivalence relation, i.e. it is *reflexive*, *symmetric* and *transitive*.

Reflexive An expression A must be coreferential to itself.

Symmetric If expression A is coreferential to expression B, then A and B both refer to the same entity and thus B is also coreferential to A.

Transitive Given a pair of co-referring expressions A and B, if there exists an expression C such that C is coreferential to B, then C is also coreferential to A, as the three expressions all refer to the same entity.

We can think of a document as a graph and the expressions in the document are the nodes of the graph. If two expressions are coreferential, we connect the corresponding nodes via a non-directed edge. In this way, the coreference relations between expressions in a document can be described by a non-directed graph. Nodes occurring in a connected subgraph are coreferential to each other.

2.1.3 Coreference and Anaphora

In the linguistic literature, one term closely related to *coreference* is *anaphora*. As in the definition by Halliday and Hasan (1976):

> **Anaphora** is cohesion which points back to some previous item.

The "pointing back" is called an *anaphor* and the previous mentioned expression to which it refers is its *antecedent*. For example, in (Eg 2.1), [$_5$ The company] refers back to [$_1$ Microsoft Corp.]. Therefore, [$_5$ The company] is an anaphor with [$_1$ Microsoft Corp.] being its antecedent. Similarly, [$_2$ its] is an anaphor which refers back to the antecedent [$_1$ Microsoft Corp.].

According to the definitions of coreference and anaphora, an anaphor and its antecedent should be coreferential to each other[1]. However, it should be noted that *anaphora* should not be confused with *coreference*; The former is a non-symmetrical and non-transitive relation that has to be interpreted in context, while the latter, as discussed in the previous subsection, is an equivalence relation held on any two expressions that have the same referent, regardless of their contexts.

2.1.4 Coreference Phenomena in Discourse

There are many ways that two expressions in a text refer to the same entity in the world. Here we provide some coreference phenomena grouping by the types of the anaphoric expressions, which can be often seen in various genres (The examples are adopted from documents in the newswire and the biomedical domains).

[1]Exception exists that an anaphor and its antecedent are not coreferential, for example, in identity-of-sense anaphora ("*The man*$_1$ who gave *his*$_1$ *paycheck*$_2$ to *his*$_1$ wife was wiser than *the man*$_3$ who gave *it*$_2$ to *his*$_3$ *mistress*", "If you do not like to attend *a tutorial*$_1$ in the morning, you can go for *the afternoon one*$_1$") and bound anaphora ("*Every participant*$_1$ had to present *his*$_2$ paper") (Mitkov et al., 2000).

- **Pronouns**

  One common coreference relation is held between pronominal anaphors and their antecedents.

  (**Eg 2.2**) *The Post may not survive long enough for* ***Mr. Murdoch*** *to get the necessary approval to buy the paper, which* ***he*** *owned from 1976 to 1988.*

  (**Eg 2.3**) ***The Thy-1 gene promoter*** *resembles a " housekeeping " promoter.* ***It*** *can only be activated in a tissue-specific manner by elements that lie downstream of the initiation site.*

- **Demonstrative and Definite Description**

  Demonstrative descriptions (i.e., noun phrases beginning with a demonstrative determiner like *this/that*) and definite descriptions (i.e., noun phrases beginning with *the*) can both be used as anaphors that refer back to an expression already mentioned in the discourse[2]. Coreference can be held between such anaphoric descriptions and their antecedents, usually realized by repetition of the head word, or by substitution with semantically close words, e.g., synonyms or hyponyms (known as "bridging")[3]. For example:

  (**Eg 2.4**) *Arrow Investments Inc., in* ***December*** *agreed to purchase $ 25 million of QVC stock in a privately negotiated transaction. At* ***that time****, it was announced that...*

  (**Eg 2.5**) *When U937 cells were infected with HIV-1, no induction of* ***NF-KB factor*** *was detected, whereas high level of progeny virions was produced, suggesting that* ***this factor*** *was not required for viral replication.*

[2]In linguistics, demonstrative description and definite description with the anaphoric use are subject to slightly different conditions (Roberts, 2002).

[3]In (Poesio and Vieira, 1998) and (Vieira and Poesio, 2000), the authors give a very comprehensive corpus-based investigation of the definite description use.

(**Eg 2.6**) *His appointment is a strong sign that* ***IBM****'s new chairman plans a similar strategy at* ***the wounded computer giant****.*

(**Eg 2.7**) *We generated* ***transgenic mice carrying the human IRF-1 gene*** *linked to the human immunoglobulin heavy-chain enhancer. In* ***the transgenic mice****, all the lymphoid tissues examined showed ...*

- **Names and Named Entities**

  Coreference can be held between names (or named-entities) and their preceding antecedents, realized by name alias, appositions and so on. For example:

  (**Eg 2.8**) *The production of* ***human immunodeficiency virus type 1*** *progeny was followed in the U937 promonocytic cell line. .... In nuclear extracts from monocytes or macrophages, induction of NF-KB occurred only if the cells were previously infected with* ***HIV-1****.*

  (**Eg 2.9**) *Footprinting analysis revealed that* ***the identical sequence CCG-AAACTGAAAAGG****,* ***designated E6****, was protected by nuclear extracts*

2.2 Coreference Resolution

2.2.1 Coreference Resolution Task

In a text, an expression and more than one of the preceding (or following) noun phrases may be coreferential and thus form a *coreferential chain* (Mitkov, 2002). The task of coreference resolution is to identify coreferential expressions and find out all the coreferential chains contained in a text. Considering the example text in Eg 2.1, the correct coreference resolution result should include two coreferential chains as below:

- "[$_1$ Microsoft Corp.] - [$_2$ its] - [$_5$ The company]"
- "[$_3$ new CEO] - [$_6$ he]"

One task related to coreference resolution is *anaphor resolution*, which refers to the process of determining the correct antecedents for given anaphors. In coreference resolution, the anaphoricity of encountered expressions is unknown. This requires that a coreference resolution system not only can identify the antecedent for an anaphor, but also can refrain from resolving a non-anaphor. Hence, the task of coreference resolution is a bigger challenge than the task of anaphora resolution.

Coreference resolution is very important for effective processing of natural language texts, and plays an important role in many NLP applications such as machine translation (Wada, 1990; Chen, 1992; Saggion and Carvalho, 1994; Mitkov et al., 1997), question answering (Morton, 1999; Breck et al., 1999), text summarization (Boguraev and Kennedy, 1997; Baldwin and Morton, 1998; Azzam et al., 1999), information extraction (Srivinas and Baldwin, 1996; Gaizauskas and Humphreys, 1997; Kameyama, 1997) and so on.

In MT, the translation of pronouns is in some cases difficult without accurate resolution of the pronouns. A pronominal anaphor in the source language could be elliptically omitted in the target language (e.g., Spanish, Italian, Japanese, Korean), or could be translated to two or more possible words (Chinese, Korean), depending on the syntactic information and semantic class of the noun to which the pronoun refers (Mitkov et al., 1995; Mitkov and Schmidt, 1998). For example, in English-Chinese translation, a pronoun "they" can be translated to:

他们，她们，它们

if the antecedent is male, female or non-human respectively.

Coreference resolution is also key to question answering. In a discourse, one entity is very likely mentioned multiple times. The full information related to the entity cannot easily be figured out, unless the mentions of the entity scattered in the text are identified. As an example, considering a sentence in a text "He is the CEO of Microsoft", the name information of the person who is the CEO of Microsoft only appears in the previous mention. If the co-referring expression of *He* fails to be determined successfully, a QA system is prone to miss the correct answer when asked "Who is the CEO of Microsoft?".

Accurate coreference resolution is especially important for information extraction. To fill the template and further merge different templates should have the knowledge whether elements within or across the templates are referents of the same entity, which heavily relies on the results of coreference resolution.

Due to its importance, coreference resolution has received more and more research interest in recent years. Particularly, in the most recent two DARPA Message Understanding Conferences, MUC6 (MUC-6, 1995) and MUC7 (MUC-7, 1998), coreference resolution is defined as a separate information extraction subtask, bridging the named-entity recognition task and template element task[4]. In the Automatic Content Extraction Program (ACE, 2000) which aims to develop automatic content extraction technology to support automatic processing of source languages, coreference resolution has also been emphasized in the subtask of entity-mention detection.

2.2.2 Evaluation of Coreference Resolution

Scoring the performance of a coreference resolution system is an important aspect of coreference resolution study, which provides a measure of how well the system performs and determines directions for further improvements. So far, several different

[4]The Information Extraction task in MUCs includes Named Entity Recognition, Coreference Resolution, Template Elements Filling, Template Relation Filling and Scenario Templates Filling.

scoring schemas have been proposed for coreference evaluation (Vilain et al., 1995; Bagga and Baldwin, 1998; Popescu-Belis and Robba, 1998; Luo, 2005).

One simple scheme adopting recall and precision is to evaluate the ability of a coreference resolution system in resolving the anaphors occurring in texts. In such a scheme the *Recall* and *Precision* rates are computed as follows:

$$\text{Recall} = \frac{\text{the number of anaphors resolved correctly}}{\text{the number of anaphors}} \tag{2.2}$$

$$\text{Precision} = \frac{\text{the number of anaphors resolved correctly}}{\text{the number of anaphors upon which resolution is attempted}} \tag{2.3}$$

And *F-measure* is the harmonic mean of *Recall* and *Precision*:

$$\text{F-measure} = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{2.4}$$

For some tasks that focus on anaphora resolution where every anaphor is to be resolved, the recall rate is identical to the precision. In such cases the term *Success* is used instead of *Recall* and *Precision*.

However, the above definitions of recall and precision do not capture the nature of coreference relation. In coreference resolution, even though a system fails to determine the coreference between two expressions, the relationship can still be recovered by virtue of its transitivity. For example, see the sentences in (Eg 2.1) which we repeat here:

[$_1$ Microsoft Corp.] announced [$_3$ [$_2$ its] new CEO] [$_4$ yesterday]. [$_5$ The company] said [$_6$ he] will ...

In the above text, the coreference relationship between $[_2$ its] and $[_5$ The company] may not be easily figured out. However, due to the transitivity, the correct coreferential chain can still be generated on condition that the reference between "$[_1$ Microsoft Corp.] - $[_2$ its]" and "$[_1$ Microsoft Corp.] - $[_5$ The company]" are successfully identified. That is, we can obtain a correct coreference resolution result even though not all the coreferential pairs in the discourse have been discovered. Therefore, a recall and precision rate calculated based on eq. 2.2 and eq. 2.3 is probably inaccurate to reflect the actual performance of a coreference resolution system.

In MUC-6 and MUC-7, a scoring algorithm by Vilain et al. (1995) was adopted to evaluate the performance of coreference resolution systems. Unlike the above mentioned scheme, Vilain et al.'s algorithm focuses on whether the coreference chains are found correctly. When the algorithm is run, it reads in a text which has been annotated with the coreference information (*key*), and compares a file output by a coreference resolution system (*response*).

In the algorithm, a coreferential chain is referred to as an equivalence class. Suppose S is the equivalence class set in the *key*, and $R_1,\ldots,R_m$ are equivalence classes generated by the response. To compute the recall, the following functions are defined:

- $p(S)$ is a partition of S relative to the response. Each subset of S in the partition is formed by intersecting S and those response set R_i that overlap S. For example, given $S = \{A\ B\ C\ D\}$ and the response $< A - B >$, the relative partition $p(S)$ is $\{A\ B\}\{C\}\{D\}$.

- $c(S)$ is the minimal number of correct links necessary to generate S, which is one less that the cardinality of S, i.e., $c(S) = |S| - 1$;

- $m(S)$ is the number of links necessary to reunite any components of the $p(S)$ partition, which is simply one fewer than the number of elements of $p(S)$; that is, $m(S) = |p(S)| - 1$;

For a single equivalence class S in the key. The recall error is the number of missing links divided by the number of correct links, i.e. $m(S)/c(S)$. Thus the recall for S is:

$$\begin{aligned} 1 - \frac{m(S)}{c(S)} &\Rightarrow 1 - \frac{|p(S)| - 1}{|S| - 1} \\ &\Rightarrow \frac{(|S| - 1) - (|p(S)| - 1)}{|S| - 1} \\ &\Rightarrow \frac{|S| - |p(S)|}{|S| - 1} \end{aligned} \tag{2.5}$$

Extending this measure from a single key equivalence class to an entire set simply requires summing over the key equivalence classes. That is,

$$Recall = \frac{\sum (|S_i| - |p(S_i)|)}{\sum (|Si| - 1)} \tag{2.6}$$

Precision is computed by switching the roles of the key and response in the above formulation.

As an example, given a text segment containing 12 NPs, denoted by 1,2,...,10, 11, 12. Suppose the key and response are:

Key: $\{1, 2, 3\}$ $\{4, 5, 6, 7, 8\}$ $\{9, 10, 11, 12\}$

Response: $\{1, 2, 3\}$ $\{4, 5, 6, 7, 8, 9, 10, 11, 12\}$

The partitions p(S1), p(S2) and p(S3) will be $[\{1, 2, 3\}]$,$[\{4, 5, 6, 7, 8\}]$ and $[\{9, 10, 11, 12\}]$ respectively. Thus the recall is

$$Recall = \frac{(3 - 1) + (5 - 1) + (4 - 1)}{(3 - 1) + (5 - 1) + (4 - 1)} = 9/9 = 100\%$$

Reversing the roles of the key and the response, the S1 and S2 will be $\{1, 2, 3\}$ and $\{4, 5, 6, 7, 8, 9, 10, 11, 12\}$, and the partitions p(S1) and p(S2) are $[\{1, 2, 3\}]$ and $[\{4, 5, 6, 7, 8\}, \{9, 10, 11, 12\}]$. Thus the precision can be calculated:

$$Precision = \frac{(3 - 1) + (9 - 2)}{(3 - 1) + (9 - 1)} = 9/10 = 90\%$$

Vilain et al. (1995)'s evaluation scheme has several shortcomings. First, the scheme overlooks the singletons, the entity that occurs in a coreferential chain containing only one element (Bagga and Baldwin, 1998). Second, it considers all errors to be equal and cannot distinguish the resolution results with different qualities (Bagga and Baldwin, 1998). Third, the scheme is "maximally indulgent" in that it just computes the minimal number of errors that may be attributed to the resolution system, which would likely lead to an irrelevant figure in some cases (Popescu-Belis and Robba, 1998). To deal with these shortcomings, several more advanced evaluating schemes have been proposed (Bagga and Baldwin, 1998; Popescu-Belis and Robba, 1998; Luo, 2005). However, Vilain et al. (1995)'s scheme is still widely employed in most coreference resolution systems so far. And for better comparison with others' work, in our study we will also adopt this scheme to do the coreference resolution evaluation.

Chapter 3

Literature Review

Coreference resolution has long been recognized as an important and difficult problem by researchers in linguistics, philosophy, psychology and computer science. This chapter will give a review of literature on the research of coreference resolution, which is organized in a way which reflects the trend of the research in this field. The chapter begins with the traditional non-learning based work which uses the early knowledge-rich approaches that heavily rely on semantics, syntax or discourse knowledge, and more recent knowledge-poor approaches. Then it presents the learning-based work which uses unsupervised, supervised and semi-supervised learning approaches.

3.1 Non-Learning Based Approaches

3.1.1 Knowledge-Rich Approaches

Wilks (1975)

Much early work on coreference resolution relies heavily on semantic knowledge. One representative of such work was *Preference Semantics*, which was proposed by Wilks (Wilks, 1973; Wilks, 1975) to determine the antecedents of pronouns. Consider

the following sentence:

(**Eg 3.1**) Give [$_1$ the bananas] to [$_2$ the monkeys] although [$_3$ they] are not ripe, because [$_4$ they] are very hungry.

Here [$_4$ they] can be interpreted correctly based on the semantic knowledge that the monkeys belong to the concept of "Animate" and only elements under this concept are likely to be hungry. Similarly, [$_3$ they] can be correctly resolved given the knowledge that only bananas, as a "Plant", are likely to be ripe.

Wilks' algorithm takes four levels of resolution depending on the type of anaphora and the mechanism needed to resolve it. The lowest level, type "A" anaphora, uses only the above mentioned Preference Semantics. If a noun phrase fails to find a unique antecedent for the anaphor, the following levels are applied in turn:

- Type "B": Analytic inference
- Type "C": Inference using real-world knowledge beyond the simple word meaning
- Type "D": focus of attention

The shortcoming of Preference Semantics, and other semantics knowledge based approach like Deep Semantic Processing (DSP) by Charniak (Charniak, 1972), is that an enormous amount of common-sense knowledge and a large number of inferences may be required for a very simple scenario, even though many restrictions might be imposed to constrain the amount of knowledge and inferencing (as in the "Blocks World" proposed by Winograd (1972)).

Hobbs (1976)

In addition to semantic knowledge, syntactic knowledge was also widely employed in the early work. Hobbs (1976), for example, proposed a syntax-based algorithm to resolve the reference of pronouns. Hobbs' algorithm works by searching the parse tree of input sentences. Specifically, the algorithm processes one sentence at a time, using a left-to-right breadth-first searching strategy. It first checks the current sentence where the pronoun occurs. The first NP that meets the syntactic constraints, like number and gender agreements, is selected as the antecedent. If the antecedent is not found in the current sentence, the algorithm traverses the trees of previous sentences in the text in reverse chronological order until an acceptable antecedent is found.

In Hobbs' algorithm, the salience of an antecedent candidate is determined by the distance between the candidate and the pronoun in the parse trees. Specifically, it prefers candidates within the same sentence and especially those closer to the pronoun in the sentence. The left-to-right breadth-first searching strategy suggests that the algorithm also prefers candidates in the subject position.

Although the algorithm does not work in all cases, the results of an examination on several hundred examples from an archaeology book, an Arthur Hailey novel and a copy of Newsweek showed that it performed remarkably well (with a success rate of 88%) in pronoun resolution. The performance was comparable with more recent sophisticated methods (Walker, 1989).

Compared with the semantics-based approaches, Hobbs' algorithm is computationally cheap. However, this algorithm is based on the assumption that one could produce the correct syntactic structure of the input sentences (Hirst, 1981). Like other syntax-based work (Bobrow, 1964; Winograd, 1972; Woods et al., 1972), the performance of the algorithm heavily depends on the results of the pre-processing parsing module.

Brennan et al. (1987)

While the syntactic constraints or semantic selectional restriction can deal with some types of reference, they cannot handle reference in general if the discourse structure is not taken into account. Indeed, discourse-based anaphora resolution is of continuing interest to attract many researchers. So far, many discourse theories have been proposed including discourse-cohesion (Lockman, 1978), concept activatedness (Kantor, 1977), logical formalism (Webber, 1978), centering or focus (Grosz, 1977; Sidner, 1978), and so on. Among them, the theory of centering receives the most interest.

Focus or centering theory provides a way to track down the focus of attention of discourse participants. A candidate which is the focus is most salient to be referred to by the current pronominal anaphor or definite description. In (Grosz, 1977; Grosz et al., 1983) and their more recent work (Grosz et al., 1995), the authors studied the representing, searching, and maintaining of the focus of attention and evaluated its effect on the resolution of definite descriptions. Such a framework was further applied to pronoun resolution by Brennan et al. (1987) (BFP).

Centering theory asserts that the discourse structure has three components:

1. the linguistic structure, which is the structure of the sequence of utterances;

2. the intentional structure, which is a structure of discourse-relevant purposes;

3. the attentional state, which is the state of focus.

The attentional state models the discourse participants' focus of attention determined by the other two structures at any one time.

The centering model contains two data structures for tracking the local focus of a sentence (utterance): the *backward-looking center* (Cb) and the list of the *forward-looking centers* (Cf). Given a discourse, each utterance U_i is assigned a list of forward-looking centers $Cf(U_i)$, and a unique backward-looking center $Cb(U_i)$. The elements

of $Cf(U_i)$ are ranked (commonly based on the grammatical relations, e.g. subject $\succ$ direct object $\succ$ indirect object) and the highest ranked one is called the preferred center (Cp). The model has the constraints that each element of $Cf(U_i)$ must be realized in U_i, and $Cb(U_i)$ is the highest ranked element of $Cf(U_{i-1})$ that is realized in U_i.

In BFP, the following centering transition states are defined:

	$Cb(U_i) = Cb(U_{i-1})$	$Cb(U_i) \neq Cb(U_{i-1})$
$Cb(U_i) = Cp(U_i)$	Continuing	Smooth-Shift
$Cb(U_i) \neq Cp(U_i)$	Retain	Rough-Shift

And two rules on the movement of center are proposed:

Rule1 If some element of $Cf(U_{i-1})$ is realized as a pronoun in U_i, then so is $Cb(U_i)$

Rule2 Transition states are ordered. Specifically, Continuing $\succ$ Retain $\succ$ Smooth-Shift $\succ$ Rough-Shift.

Finally, the following three steps are taken to resolve the pronominal anaphors:

1. Generate all possible $Cb - Cf$ combinations.
2. Filter the $< Cb, Cf >$ pairs by the contra-indexing and centering rules.
3. Rank the remaining pairs according to the transition orderings.

Walker (1989) evaluated BFP on three small data sets, which was compared with Hobbs' algorithm. The results indicated that Hobbs's algorithm outperformed BFP over a news domain (80% vs 79%) and a task domain (51% vs 49%).

One problem with BFP is that it makes no provision for incremental resolution of pronouns (Kehler, 1997a). Motivated by BFP's limitation, several algorithms were

proposed like S-List algorithm (Strube, 1998; Strube and Hahn, 1999) and LRC (Left-Right Centering) algorithm (Tetreault, 1999). Tetreault (2001) gives a corpus-based evaluation of these centering-based algorithms.

3.1.2 Knowledge-Poor Approaches

Unlike the above mentioned semantics, syntax or discourse based approaches, knowledge-poor approaches do not rely on the specific knowledge to make reference determination. Instead, they make use of various sources of shallow knowledge that is computationally cheap and more domain-independent.

Baldwin (1997)

Baldwin (1997) proposed a pronoun resolution system CogNIAC, which focuses on resolving the set of anaphors that do not require general world knowledge or sophisticated linguistic processing. The information used in the system only includes sentence detection, part-of-speech tagging, gender/number identification, and partial parse trees.

In CogNIAC, the resolution is run on a set of heuristic rules, which take forms like:

- "If there is a single possible antecedent i in the read-in portion of the entire discourse, then pick i as the antecedent"
- "Pick nearest possible antecedent in read-in portion of current sentence if the anaphor is a reflexive pronoun"
- "If there is a single possible antecedent i in the prior sentence and the read-in portion of the current sentence, then pick i as the antecedent"

- "If the anaphor is a possessive pronoun and there is a single exact string match i of the possessive in the prior sentence, then pick i as the antecedent"
- "If there is a single possible antecedent in the read-in portion of the current sentence , then pick i as the antecedent"
- "If the subject of the prior sentence contains a single possible antecedent i, the anaphor is the subject of the current sentence, then pick i as the antecedent".

For each pronoun encountered, the above rules are applied in order until a given rule can lead to the determination of an antecedent. If no rules can resolve the pronoun, then it is left unresolved.

CogNIAC reported 92% precision and 64% recall on 298 third person pronouns. It also reported 75% recall and 73% precision when tested on the all pronouns in MUC-6.

The advantage of rules is that they can be easily deployed and lead to a high performance for a specified domain. For this reason, rule-based approaches are widely used in many practical coreference resolution systems (e.g.,
Williams et al. (1996)). Recently, Zhou and Su (2004) proposed a more sophisticated rule-based system for coreference resolution. Their system discriminated and used separate rules (called *agents* in their work) to handle different types of coreference phenomena (e.g., pronouns, definite nouns, bare nouns, etc). They reported a high coreference resolution performance for the MUC-6 and MUC-7 data set, achieving precision as high as 80% with recall in the range 55% - 65%.

Lappin and Leass (1994)

Different from rule-based algorithms as introduced in the previous subsection, salience-based approaches use a set of salience factors to represent the multiple knowledge

considered for reference resolution. Each factor has a weight reflecting the relative importance of the particular knowledge in the reference determination.

In (Lappin and Leass, 1994), the authors proposed such a salience-based algorithm, RAP (Resolution of Anaphora Procedure), for identifying the antecedents of third person pronouns. The algorithm relies on salience measures derived from syntactic structure and a simple dynamic model of attentional state to select the antecedent of a pronoun from a list of candidates. It does not employ semantic conditions (beyond those implicit in grammatical number and gender agreement) or real-world knowledge in choosing among the candidates. Neither does it model focus or global discourse structure.

The kernel part of RAP algorithm is the procedure for assigning values to several salience parameters (grammatical role, parallelism of grammatical roles, frequency of mention, proximity, and sentence recency) for a noun phrase. The algorithm assigns salience weights based on the following preference rules (i) subject over non-subject NPs,(ii) direct objects over other complements,(iii) arguments of a verb over adjuncts, and objects of a prepositional phrase over adjuncts of the verb, (iv) head nouns over complements of head nouns. For instance, the salience factor *Sentence recency* is assigned the initial weight of 100, while *Subject emphasis* is 80.

All the discourse referents evoked by a new sentence are tested to see if the salience factors could apply. All the salience factors that have been assigned prior to the new sentence will have their weights degraded by a factor of two.

Given a third-person pronoun, a list of possible antecedent candidates are created, which contains the most recent discourse referent of each coreferential chain. The salience of each candidate is computed as the sum of the salience values of the elements in its current chain. A decision procedure is incorporated to select the preferred antecedent from the set of candidates based on the salience values. The candidate with the highest salience is the most likely to be chosen as the antecedent.

The authors tested the algorithm extensively on computer manual texts and conducted a blind test on a manual text containing 360 pronoun occurrences. The algorithm successfully identified the antecedent in 86% of the cases, which performed better than Hobbs' algorithm.

Later, Kennedy and Boguraev (1996) gave a modified and extended version of Lappin and Leass' approach. Their system does not require in-depth and full syntactic parsing but works on POS tagging and grammatical functions of lexical items. They reported 75% success, on a random selection of genres including press releases, magazine articles or web pages.

Another remarkable extension of RAP was made by Mitkov (1998), who proposed and investigated a comprehensive list of different salience factors (called indicators in Mitkov (1998)'s work). In his algorithm, candidates are assigned a score (2, 1, 0, -1) for each salience indicator; the candidate with the highest aggregate score is proposed as the antecedent.

In Mitkov's work, the antecedent indicators have been identified on the basis of empirical studies and the majority of them are genre-independent. The indicators are related to salience (e.g., definiteness, indefiniteness, giveness, lexical reiteration), structural matches (e.g., collocation, sequential structure), referential distance or to preference of terms. They can be "impeding" (non-PP NPs, definiteness/indefiniteness), assigning negative scores to candidates or "boosting" (the rest), assigning positive scores. For instance, the indicator "definiteness" considers definite noun phrases better than the indefinite ones to be the antecedent, and therefore, indefinite noun phrases are penalized by the negative score of -1. Noun phrases in previous sentences and clauses representing the "given information" (theme) are deemed good candidates are thus assigned a score of 1.

The approach was evaluated on a corpus of technical manuals containing 223 pronouns, and achieved a success rate of 89.7%.

Salience-based approaches have been widely seen in the work on coreference resolution. The strength of this kind of approach is that knowledge for reference can be encoded in terms of salience factors. That makes it possible to apply machine learning methods to investigate the impact of different sources of knowledge o coreference resolution. In the next section, we will have a review of the work that uses learning-based approaches to resolve coreference.

3.2 Learning-based Approaches

The work described in the previous section is non-learning based. That is, those coreference resolution algorithms are based on hand-crafted constraints and preference heuristics. Consider the searching order in Hobbs' algorithm, the ranking of forward-looking centers in the centering models, the resolution rules in the rule-based algorithms and the weights of the salience indicators in the salience-based algorithms. All of them are manually designed, which heavily relies on experts' knowledge and lacks adaptivity across domains. Consequently, recent years have seen more and more corpus-based approaches that employ machine-learning techniques to automatically discover the regularities for coreference resolution.

Machine-learning (ML) is an AI field that studies how to learn the connection between features of the examples and a specified target concept. In the past couple of decades, the blending of ML and NLP becomes increasingly common with the expanding availability of large corpora. The empirical methods let learning algorithms acquire the knowledge from available data, and thus reduce the dependence of manually embedding knowledge into NLP systems. As a result, it is not uncommon nowadays to see that most well known ML techniques have been applied to almost every possible NLP task, which also includes coreference resolution.

In this section, we will describe some of the previous work that applies machine

learning, either unsupervised, supervised or weakly-supervised, to the coreference resolution problem.

3.2.1 Unsupervised-Learning Based Approaches

Cardie and Wagstaff (1999)

Cardie and Wagstaff (1999) proposed an unsupervised clustering-based approach to coreference resolution. Their approach views coreference as a partitioning or clustering the noun phrases, based on the fact that coreference is an equivalence relation. The assumption of the approach is that all of the noun phrases used to describe a specific concept are "close" in conceptual distance. Therefore, given a method for measuring the distance between two noun phrases, a clustering algorithm can be utilized to group noun phrases: Noun phrases with distance greater than a clustering radius r are not placed into the same partition and so are not considered coreferential.

The metric for the distance between two NPs is defined as follows:

$$dist(NP_i, NP_j) = \sum_{f \in F} W_f * incompatibility_f(NP_i, NP_j) \tag{3.1}$$

where F corresponds to the feature set of a noun phrase; $incompatibility_f$ is a function that returns a value between 0 and 1 inclusive and indicates the degree of incompatibility of a feature f for NP_i and NP_j; and W_f denotes the relative importance of compatibility with regard to f. In their algorithm, eleven incompatibility functions associated with various manually specified weights are defined. Some of the incompatibility functions, e.g., *semantic-class-mismatch* and *gender-mismatch*, are assigned a weight of ∞ indicating that NP_i and NP_i are incompatible regardless of the other incompatibility functions.

The algorithm starts at the end of the document and works backwards. Initially, every noun phrase NP_i is marked as belonging to its own cluster, C_i. During resolution, for a noun phrase NP_j encountered, each preceding noun phrase NP_i is to

be considered. If the distance between NP_i and NP_j is below the specified radius threshold r, the clusters where the two NPs reside are merged, on condition that no NP pair across the two clusters is incompatible. Such a process continues until the beginning of the document is reached.

In essence, this approach is similar to the salience-based approaches (features are in fact the salience factors), but uses a different NP clustering strategy. The approach was evaluated on MUC-6 data set, and obtained 48.8% recall and 57.4% precision.

Bean and Riloff (2004)

Bean and Riloff (2004) presented a coreference resolution system called BABAR that incorporates the contextual-role knowledge to identify antecedents for given anaphors. BABAR employs information extraction techniques to represent and learn role relation, and uses unsupervised learning to acquire this knowledge from plain texts without the need for annotated training data.

The first step of the learning process of BABAR is to generate a set of seeds, i.e., the anaphor and antecedent pairs that can be easily and reliably resolved. Then, it applies the AutoSlog system (Riloff, 1996) to the un-annotated training texts, which generates a large set of caseframes coupled with a list of extracted noun phrases. To perform coreference resolution, BABAR utilizes the following contextual role knowledge derived from the caseframe data :

- The caseframe network: An anaphor and a candidate may be coreferential if the caseframe where they reside co-occurs.
- Lexical caseframe expectations: An anaphor and a candidate may be coreferential if the anaphor and the candidate are substitutable for each other in their caseframes.

- Semantic caseframe expectations: An anaphor and a candidate may be coreferential if they are substitutable for one another in their caseframes, based on their semantic classes.

The above knowledge, plus additional seven sources of general knowledge like gender/number/semantic matching, distance, recency, scoping and so on, is combined together to resolve coreference, using a Dempster-Shafer decision model (Stefik, 1995). BABAR was run for definite NP anaphors and pronominal anaphors of MUC-6 corpus, on the terrorism and disaster domains. The seven general knowledge sources led to 42-50% recall for both domains, with around 80% precision. The three sources of unsupervised-learned contextual knowledge could further bring up to 15% gain in recall for the resolution of pronominal anaphors.

3.2.2 Supervised-Learning Based Approaches

Ge et al. (1998)

Ge et al. (1998) proposed a probabilistic model to resolve pronominal anaphors. Their model considers several training features such as sentence distance, syntactic role, mentioned times, etc. For each anaphor p and a list of antecedent candidates $\vec{W}$, the probability that a candidate a is the antecedent of p is:

$$f(a,p) = P(A(p) = a|p, h, \vec{W}, t, l, sp, \vec{d}, \vec{M}) \tag{3.2}$$

where $A(p)$ is a random variable denoting the referent of the pronoun p and a is the proposed antecedent. In the conditioning event, h is the head constituent above p; $\vec{W}$ is the list of antecedent candidates; t is the phrase type of the proposed antecedent; l is the type of the head constituent; sp describes the syntactic structure in which p appears; $\vec{d}$ specifies the distance of each antecedent from p and $\vec{M}$ is the number of times the referent is mentioned. Here the probability $f(a,p)$ can be regarded as

the score of an antecedent candidate. The candidate with the highest probability is selected as p's antecedent, that is

$$F(p) = \arg\max_{a \in \vec{W}} f(a, p) \tag{3.3}$$

In Ge et al.'s work, the above probability formula has been simplified based on a set of independence assumptions, and finally arrives at the following equation for computing the probability of each proposed antecedent:

$$f(a, p) \approx \frac{P(d_H|a)P(p|w_a)P(w_a|h, t, l)P(a|m_a)}{p(w_a|t)} \tag{3.4}$$

where d_H is the Hobbs distance (Hobbs, 1978) which combines sp and d_a. The components in the above equation can be estimated in a reasonable fashion. $P(d_H|a)$ can be obtained by running Hobbs' algorithm on the training data. Since the training corpus is tagged with reference information, the probability $P(p|w_a)$ can be easily calculated. In building a statistical parser for the Penn-Tree-bank various statistics have been collected, two of which are $P(w_a|h, t, l)$ and $P(w_a|t, l)$. The corpus also contains referent's repetition information, from which $P(a|m_a)$ can be directly computed.

The experiments on a data set consisting of 93,931 words and 2447 pronouns showed the algorithm can achieve 82.9% success rate.

Kehler (1997b) also proposed a probability-based algorithm to do coreference resolution. His algorithm works by assigning a probability distribution, based on the maximum entropy modelling, to the possible sets of coreference relationships among noun phrase entity templates.

Connolly et al. (1997)

Ge et al. (1998)'s algorithm is in fact based on a Bayesian learning model in which the probability of a candidate to be the antecedent is calculated based on the prior statistics learned from the annotated data. Such a model, however, is complicated to

represent and requires that the incorporated features be independent of each other. In practice, a more simple and efficient solution to apply supervised learning techniques is to view coreference resolution as a classification problem, and learn a classifier to do the job.

Connolly et al. (1997) proposed a trainable learning approach to anaphora resolution. Their approach decomposes the candidate selection problem into a binary classification problem. In the approach, a training or testing instance is a feature vector extracted from a 3-tuple, the anaphor and two antecedent candidates. The features describe the properties of the anaphor and the two candidates (lexical type and grammatical role), as well as their relationships (recency and number/gender/semantic Agreement). A label is assigned to each instance, indicating which of the two candidates is more likely the antecedent of the anaphor.

Based on the feature vectors generated, a classifier can be trained using a machine learning algorithm. It is supposed to judge, between two antecedent candidates of an anaphor, which one is better than the other for the antecedent.

During resolution, an encountered anaphor and two of its candidates are paired as an instance associated with a specified feature vector, and then presented to the classifier which then returns a class label indicating the preference between the two candidates.

For the final antecedent selection, candidates are compared in sequence either forwards or backwards. The "losing" candidate is discarded and the winner is compared with the next candidate. The process continues until every candidate has been examined, and the winner from the last triple is chosen as the antecedent.

Connolly et al. (1997)'s work was evaluated on a corpus of 80 news-agency articles. They reported a success of 55.3% for pronominal anaphora resolution, and 37.4% for definite anaphora resolution. The results were better than their handcrafted algorithm which gave 51.6% and 25.7% success for the two types of anaphora,

respectively.

Later, Iida et al. (2003) also used a similar algorithm to do Japanese zero-anaphora resolution, and reported a success rate of around 70%.

Soon et al. (2001)

A more common representation seen in learning-based coreference resolution systems is based on a pair of anaphor and one candidate (Aone and Bennett, 1995; McCarthy and Lehnert, 1995; Soon et al., 2001; Ng and Cardie, 2002b; Strube and Mueller, 2003). Here we would like to describe the system by Soon et al. (2001) as the paradigm.

The main idea of Soon et al. (2001)'s approach is to recast coreference resolution as a binary classification problem. Specifically, the classification is done on an anaphor and antecedent candidate pair to test whether they are coreferential or not.

During training, a set of instances is generated for each anaphor in an annotated text. A training instance is formed by the anaphor and one of its antecedent candidates, which are restricted to the NPs which occur between the anaphor and its immediate antecedent. An instance is labelled as positive or negative based on the fact whether or not the candidate belongs to the same coreferential chain as the anaphor.

A feature vector is specified for each training instance. The features may describe not only the characteristics of the anaphor and the candidate, but also their relationships from lexical, syntactic, semantic, and positional aspects. Soon et al. (2001)'s approach uses twelve features that are highly domain-independent (see Table 3.1).

Based on the generated feature vectors, a classifier is trained using the C5 learning algorithm. During resolution, for each new anaphor, a test instance is formed by pairing the anaphor and one antecedent candidate. The test instance is presented to the classifier, which then returns a positive or negative class label with a confidence

Is the anaphor a pronoun?
Is the anaphor a definite noun phrase?
Is the anaphor a demonstrative pronoun?
Is the candidate a pronoun?
Do the anaphor and the candidate have the same head word?
Do the anaphor and the candidate agree in number?
Do the anaphor and the candidate agree in gender?
Do the anaphor and the candidate agree in semantics?
Are both the anaphor and the candidate proper names?
Are the anaphor and the candidate in the same appositive structure?
Is the anaphor a name alias of the candidate?
The distance between the anaphor and the candidate?

Table 3.1: Features used in the system by Soon et al. (2001)

label indicating the likelihood that the candidate is the antecedent of the anaphor.

For an anaphor, there probably exist several antecedent candidates with a positive instance label. Hence a clustering strategy should be employed to link the anaphor to a proper candidate. In (Soon et al., 2001), a "closest-first" clustering is used in which the anaphor is to be linked to the positive candidate that is closest to the anaphor in position.

Soon et al. (2001)'s approach was evaluated on both MUC-6 and MUC-7. For MUC-6, the approach obtained 58.6% recall and 67.3% precision, while for MUC-7 it obtained 56.1% recall and 65.5% precision.

3.2.3 Weakly-Supervised-Learning Based Approaches

Mueller et al. (2002)

One deficiency of supervised learning coreference resolution is the need for a set of annotated training data. Currently, annotated corpora for coreference resolution is still not large compared with those for other NLP applications. For this reason, researchers began to explore weakly supervised learning algorithms that can run with

only a small annotated data set. Mueller et al. (2002), for example, proposed to use co-training techniques to do coreference resolution.

Co-training is a meta-learning algorithm which exploits unlabelled in addition to labelled training data for classifier learning (Blum and Mitchell, 1998). A co-training classifier consists of two simple classifiers that are learned based on different subsets of the features (referred to as *views* in the literature). Initially, these classifiers are trained normally using a small set of size L of the labelled training data. The p best positive and n best negative instances returned by a classifier are added to the other classifier's training instance set. Then both classifiers are retrained and updated on their respective new data sets. In this way, the training data are gradually extended by bootstrapping. This process is repeated on the training set, until a specified iteration number is reached or all the unlabelled data has been labelled.

In Mueller et al. (2002) 's approach, they created the two views by distinguishing between features assigned to noun phrases and features assigned to the potential coreferential relation. The former view included NP-level features such as the grammatical functions, the lexical forms, gender/number/semantic agreement between the two NPs. The latter view contained coreference-level features like the position distance or minimum edit distance between the possible anaphor and the candidate.

The experiments were run on 250 German texts. The authors found that co-training would lead to considerable improvement for the resolution of definite NPs, while it seemed not every effective for other types of NPs.

Later, Ng and Cardie proposed and investigated several weakly supervised learning based algorithms that run without redundant views, like self-training (Ng and Cardie, 2003a) and EM (Ng and Cardie, 2003b). The reported results on MUC-6 and MUC-7 indicated that their algorithms are more effective than the co-training based one for the coreference resolution task.

3.3 Summary and Discussion

3.3.1 Summary of the Literature Review

In this chapter we gave a review of the previous work on coreference resolution. We organized the discussed work in an order that reflects the main trends of the research in this field. Specifically, we discriminated the work using the non-learning based approaches and the learning-based approaches.

We started with the non-learning based work. We first reviewed the early knowledge-rich approaches which rely on the semantics, syntax, and discourse knowledge. Then we introduced two knowledge-poor approaches based on rules and salience.

The semantics-based approaches, represented by (Wilks, 1975), depend heavily on world knowledge and a large number of inferences. Compared with the semantics-based approaches, syntax-based approaches like (Hobbs, 1976), are computationally cheap. However, as pointed out by Hirst (1981), syntactic knowledge by itself is inadequate for reference determination. The accuracy of the syntactic parsing results has a significant influence on the resolution performance.

Discourse-based approaches usually do coreference resolution based on the centering theory, by tracking the focus of the discourse. However, the drawback of traditional centering models like BFP (Brennan et al., 1987) is that they make no provision for incremental resolution of anaphors (Kehler, 1997a), and thus are generally difficult to be deployed for antecedent selection in practice. The modified centering algorithms such as S-List (Strube, 1998) and LRC (Tetreault, 1999) could effectively deal with this problem.

In contrast to the knowledge-rich approaches, knowledge-poor approaches can be applied to coreference resolution both reliably and efficiently. Rule-based approaches, like CogNIAC (Baldwin, 1997), are capable of leading to high performance over a specific domain. For this reason rule-based coreference resolution is popular in many

practical systems. However, the limitation of rule-based approaches is that rules usually lack adaptivity. And the management of the rules becomes a serious issue with an increasing number of rules.

Salience based approaches, like RAP (Lappin and Leass, 1994), determine the preference of candidates based on a set of salience indicators. The strength of these approaches is that different sources of knowledge can be easily represented, which provides a convenient way to incorporate various knowledge into coreference resolution.

In the second part of the chapter, we focused on machine-learning based approaches to coreference resolution, including unsupervised, supervised, and weakly-supervised approaches. Compared with the non-learning approaches, the machine learning based approaches are considered particularly attractive because they can automatically learn resolution regularities from the training data, which largely reduces the human effort in designing and implementing the resolution strategies.

Cardie and Wagstaff (1999) proposed to resolve coreference by grouping the coreferential noun phrases into separate clusters. The advantage of their approach, like other unsupervised learning based approaches described in the review, is that it does not need annotated training data. However, in their approach, the weights of the features used for the distance calculation have to be assigned manually.

Supervised learning based approaches use a training model to learn coreference resolution regularities from annotated data. The first work introduced in our review is by Ge et al. (1998). The approach determines the most preferred antecedent candidate by means of calculating the probability of each candidate, using a Bayesian learning model. One limitation of the model is that the features used in the model have to be independent of each other.

The more common practice in many coreference resolution systems is to uses classifiers to do coreference resolution. In our review, we summarized two representative

works by (Connolly et al., 1997) and (Soon et al., 2001). Both of the work recast coreference resolution as binary classification problem, but employ different classification models: In the former, the classification is done between two competing candidates to determine their preference for the antecedent. In the latter, the classification is done between a possible anaphor and its candidate to judge their coreference relationship directly. They both reported that their learning-based methods could achieve performance comparable to the manually designed heuristics.

Standing between the unsupervised and supervised paradigms, weakly supervised learning approaches work with a small set of labelled data, by expanding the training data using bootstrapping. These approaches could effectively deal with the problem that supervised learning based approaches might suffer from, i.e., the lack of a large annotated data set.

3.3.2 Comparison with Related Work

The work in this thesis is based on the twin-candidate model which is similar to that used in the work by Connolly et al. (1997) and Iida et al. (2003). However, the work in the thesis differs from these others in a number of ways:

- Both Connolly et al. (1997)'s work and Iida et al. (2003) 's work use a näive linear resolution scheme for antecedent selection, by applying the classifier to successive pairs of candidates, each time retaining the better candidate. In contrast, the work in our thesis presents various searching strategies and evaluates their effectiveness in experiments.

- Both Connolly et al. (1997)'s work and Iida et al. (2003) 's work only focus on antecedent selection for anaphora resolution. However, the basic twin-candidate model cannot judge the anaphoricity of an expression in texts, and thus cannot be directly deployed for coreference resolution. In contrast, our thesis will

investigate various strategies to make the twin-candidate model applicable to coreference resolution. In particular, the thesis will propose a modified twin-candidate model which can automatically do anaphoricity determination and antecedent selection at the same time, and thus can be reliably applied to both anaphora resolution and coreference resolution. To our knowledge, our work is the first one to do coreference resolution using the twin-candidate model.

- The previous work (Connolly et al., 1997; Iida et al., 2003) on the twin-candidate model is comparatively preliminary. In contrast, the work in our thesis will give an in-depth exploration of the twin-candidate model for the anaphora resolution and coreference resolution tasks. The thesis will cover some important issues that have never been investigated in the previous work. For example, we will have an analysis on the utilities of different types of knowledge in twin-candidate learning model. Also we will evaluate the impact of the factors that may have an influence on the resolution performance, e.g., the size of the training data. Moreover, we will compare the twin-candidate model with the single-candidate learning model that is much more commonly used.

Chapter 4

Learning Models of Coreference Resolution

The advantage of supervised-learning based approaches to coreference resolution is that reference regularities can be automatically found from the training data. Like in many other machine-learning based NLP applications, the first and the most crucial step to apply any machine learning algorithm to coreference resolution is to design the learning model of the problem. As described, to date, most of the machine learning based work on coreference resolution adopts the single-candidate paradigm, that is, an instance is composed of a possible anaphor and one antecedent candidate, with features used to describe the properties and relationships of the pair. The classification is done to determine their reference directly. However, can such a model accurately represent the coreference resolution problem? Or is there another more reasonable learning model?

This chapter will describe some possible models of the coreference resolution problem. The first part of the chapter introduces the *all-candidate model* and the *single-candidate model*, and discusses the pros and cons of these two models. Then the second part gives an introduction to the alternative twin-candidate model and ana-

lyzes its advantages over the single-candidate paradigm.

4.1 Modelling the Coreference Resolution Problem

The key step of coreference resolution is to identify the antecedent from a set of candidates[1] of a possible anaphor, or more formally, to calculate $p(ante(C_k)|ana, C_1, C_2, \ldots, C_n)$, the probability that a candidate C_k is the antecedent of the anaphor *ana* under the context of its antecedent candidates, $C_1, C_2, \ldots, C_n$.

The basic idea of machine learning based coreference resolution is to use machine learning techniques to obtain the probabilities of the candidates. Like in other learning based applications, before applying a machine learning algorithm to coreference resolution, we should first design the learning model of the problem, including

- What constitutes a training/testing instance of the problem?
- How to represent the knowledge related to the problem? What is the definition of features?
- How to use the generated classifier to solve the problem?

This section will describe two simple models of the coreference resolution problem. An example text shown in Table 4.1 will be used to demonstrate the models. In the text, each NP expression is marked with brackets and a sequence ID assigned. For simplicity, we will use NP_j to refer to the noun phrase with the sequence ID j.

In the text, we can find three coreferential chains:

[1] Generally, a candidate of a given anaphor is a noun phrase. However, in some works a candidate can be a partially found entity (Lappin and Leass, 1994; Luo et al., 2004; Yang et al., 2004b; Yang et al., 2005a). In this thesis we will only consider the candidates on the basis of noun phrases.

[$_1$ Washington Post Co.] said [$_2$ Katharine Graham] stepped down after 20 years as [$_3$ chairman] , and will be succeeded by [$_5$ [$_4$ her] son , Donald E. Graham] , [$_7$ [$_6$ the company] 's chief executive officer].
The departure of [$_8$ the matriarch of the Graham publishing empire] came as little surprise to media analysts , who have been anticipating [$_9$ Mr. Graham] 's ascent to the top spot .

Table 4.1: An example text used to demonstrate different learning models

- Chain 1

  [$_1$ Washington Post Co.]

  [$_6$ the company]

- Chain 2

  [$_2$ Katharine Graham]

  [$_3$ chairman]

  [$_4$ her]

  [$_8$ the matriarch of the Graham publishing empire]

- Chain 3

  [$_5$ her son , Donald E. Graham]

  [$_7$ the company 's chief executive officer]

  [$_9$ Mr. Graham]

4.1.1 The All-Candidate Model

One possible model of coreference resolution is *all-candidate*, which recasts the resolution problem as a multi-classification problem. The model calculates the probability that a candidate is the antecedent as follows,

Current NP	Antecedent	Instance	Class Label
NP_3	NP_2	$\boldsymbol{i}\{NP_3$, $\{NP_1,NP_2\}\}$	2
NP_4	NP_3	$\boldsymbol{i}\{NP_4$, $\{NP_1,NP_2,NP_3\}\}$	3
NP_6	NP_1	$\boldsymbol{i}\{NP_6$, $\{NP_1,\ldots,NP_5\}\}$	1
NP_7	NP_5	$\boldsymbol{i}\{NP_7$, $\{NP_1,\ldots,NP_6\}\}$	5
NP_8	NP_4	$\boldsymbol{i}\{NP_8$, $\{NP_1,\ldots,NP_7\}\}$	4
NP_9	NP_7	$\boldsymbol{i}\{NP_9$, $\{NP_1,\ldots,NP_8\}\}$	7

Table 4.2: Instances generated for the all-candidate model

$$p(\ ante(C_k) \mid ana, C_1, C_2, \ldots, C_n) \ \propto \ CF(i\{ana, C_1, C_2, \ldots, C_n\},\ K) \qquad (4.1)$$

where $i\{ana, C_1, C_2, \ldots, C_n\}$ is the instance formed by the anaphor and the candidate set, and $CF(\boldsymbol{i}, K)$ is the confidence with which the classifier returns the class label K for the instance i. The confidence is used as the approximation of the probability p. The candidate with the highest confidence value is selected as the antecedent.

To learn such a classifier, a set of training instances is created for the anaphors in the training texts. Each instance, $i\{ana, C_1, C_2, \ldots, C_n\}$, corresponds to an anaphor ana and its candidate set $C_1, C_2, \ldots, C_n$. The instance is labelled as the class K if C_k is the immediate antecedent of ana.

Consider the text in Table 4.1 as an example. The set of training instances to be generated is listed in Table 4.2.

However, such an all-candidate model will encounter many difficulties in practice.

First, as each class represents a distinct candidate in the texts, the number of classes in question is prohibitively large. Moreover, as a candidate generally does not repeat itself often in the whole data set, the number of instances associated with each class is quite sparse. As a result, it would be difficult for a machine learning algorithm to learn a classifier with an acceptable distinguishing capability.

Second, for each new NP encountered, the number of the preceding candidates is not fixed, which makes the number of features vary from instance to instance. Unfortunately, so far very few machine learning algorithms can nicely deal with the problem of variable-length features.

4.1.2 The Single-Candidate Model

The idea of the single-candidate model is to recast coreference resolution as a binary classification problem. The assumption of the model is that the probability of C_k to be the antecedent is only dependent on the anaphor *ana* and C_k, but independent of all the other candidates. That is,

$$p\ (\ ante(C_k) \mid ana, C_1, C_2, \ldots, C_n) \ = \ p\ (\ ante(C_k) \mid ana, C_k) \tag{4.2}$$

$$\propto \ CF(\boldsymbol{i}\{ana, C_k\},\ K) \tag{4.3}$$

In this way, the probability of a candidate C_k can be approximated using the classification result on the instance describing the anaphor and C_k. Thus, the model only needs to consider *one* candidate, instead of all the candidates, to do the antecedent selection.

The classifier is learned based on a set of training instances, each of which is formed by an anaphor and one antecedent candidate, $\boldsymbol{i}\{\text{ana}\ ,\ C_k\}$. A class label is assigned to an instance indicating the coreferential relationship between the pair, for example, "1" (positive) if coreferential or "0" (negative) if otherwise. After training, given a test instance, the generated classifier is supposed to return "0" or "1" with a confidence value indicating the likelihood that a candidate is the antecedent of the given NP. And the antecedent is then selected based on the confidence values of the competing candidates.

For demonstration, parts of the instances generated for the text of Table 4.1, in

Current NP	Antecedents	Instance	Class Label
NP_3	NP_2	$\boldsymbol{i}\{NP_3, NP_1\}$	0
		$\boldsymbol{i}\{NP_3, NP_2\}$	1
NP_4	NP_2,NP_3	$\boldsymbol{i}\{NP_4, NP_1\}$	0
		$\boldsymbol{i}\{NP_4, NP_2\}$	1
		$\boldsymbol{i}\{NP_4, NP_3\}$	1
...			
NP_6	NP_1	$\boldsymbol{i}\{NP_6, NP_1\}$	1
		$\boldsymbol{i}\{NP_6, NP_2\}$	0
		$\boldsymbol{i}\{NP_6, NP_3\}$	0
		$\boldsymbol{i}\{NP_6, NP_4\}$	0
		$\boldsymbol{i}\{NP_6, NP_5\}$	0
...			

Table 4.3: Instances generated for the single-candidate model

the single-candidate model, are listed in Table 4.3.

This single-candidate model is able to overcome the difficulties from which the all-candidate model suffers. The model has a limited number of the classes (i.e. "1" and "0"), rather than the large set of classes. Besides, the model has a fixed number of features, as only two elements (i.e., the anaphor and one candidate) are related to an instance. Due to these advantages, the single-candidate model has been widely adopted in coreference resolution systems, including (Soon et al., 2001) which was described in section 3.2.2 of the literature review chapter.

4.2 Problems with the Single-Candidate Model

4.2.1 Representation

As described above, the assumption behind the single-candidate model is that the reference relationship between an anaphor and a candidate is completely independent of

the other competing candidates. However, previous studies on coreference resolution have suggested that antecedent selection is often subject to the preference among the candidates (Jurafsky and Martin, 2000). For instance, Wilks' algorithm (see Section 3.1.1) prefers the candidate that is more compatible in semantics with the anaphor; Hobbs' algorithm (3.1.1) prefers the candidate that is closer to the anaphor in the syntax tree; The centering based BFP algorithm (3.1.1) prefers a subject candidate in ranking the forward-looking centers. And the RAP algorithm (3.1.2) prefers the candidate that has a higher salience value. Whether a candidate is the antecedent depends on whether it is the "best" among the candidate set, and there exists no other candidate that is preferred over it.

The single-candidate model can select the antecedent based on preference, by using the classification confidence for the candidates. Nevertheless, as the model only considers one candidate at a time, it cannot capture the preference between the candidates during training. Thus the confidence returned by the learned classifier during resolution cannot reliably represent the actual preference relationship between candidates.

(**Eg 4.1**) *Jenny bought the nice cup last week. However, yesterday she put* $[_1$ *the cup* $]$ *on* $[_2$ *a plate* $]$ *and broke* $[_3$ *it* $]$ *.*

(**Eg 4.2**) *Jenny bought* $[_4$ *a cup* $]$*, but broke* $[_5$ *it* $]$ *yesterday.*

Table 4.4: An example to demonstrate the problem with the single-candidate learning model

Consider the text in Table 4.4. Suppose we have such a preference rule for antecedent selection, i.e., *definite NP* $\succ$ *indefinite NP*. According to this, for Eg 4.1, the pronoun $[_3$ it $]$ is resolved to the definite NP $[_1$ the cup $]$ and thus two instances are generated: $\boldsymbol{i}\{[_3$ it $]$, $[_1$ the cup $]\}$ and $\boldsymbol{i}\{[_3$ it $]$, $[_2$ a plate $]\}$, labelled as positive

and negative respectively. By contrast, in Eg 4.2 there exists no definite NP in the candidate set for the anaphor [$_5$ it]. Hence the indefinite NP [$_4$ a cup], despite the low preference, is selected and a positive instance $\boldsymbol{i}${[$_5$ it] , [$_4$ a cup]} is generated. Now, we can find inconsistency in the instance set: Two instances, $\boldsymbol{i}${[$_3$ it] , [$_2$ a plate]} and $\boldsymbol{i}${[$_5$ it] , [$_4$ a cup]}, bear the same feature value (*indefinite-np*) but different class labels (negative and positive, respectively). Such inconsistency would probably lead to errors for classifier learning. For example, if in the data set the number of the sentences as in Eg 4.2 is larger than those as in Eg 4.1, we may finally obtain a classifier that gives higher confidence value to an indefinite NP than to a definite NP, which conflicts with the inherent preference rule.

To illustrate the potential errors, suppose we have a data set where candidates can be described with four exclusive features: f_1, f_2, f_3 and f_4. The ranking of candidates obeys the following preference rule:

$$C_{f_1} \succ C_{f_2} \succ C_{f_3} \succ C_{f_4},$$

where C_{f_i} $(1 \leq i \leq 4)$ represents the category of the candidates that have feature f_i.

Suppose the training data contains 1000 anaphors whose candidates are either from category C_{f_1} or category C_{f_2}, 1000 anaphors whose candidates are from C_{f_2} or C_{f_3}, and 10000 anaphors whose candidates are from C_{f_3} or C_{f_4}. As a result, the C_{f_3} candidates produce 10000 positive instances and 1000 negative instances, while the C_{f_2} candidates produce 1000 positive instances and 1000 negative instances. Thus the probability that a C_{f_3} candidate leads to a positive label is:

$$p(C_{f_3}) = \frac{10000}{1000 + 10000} = 90.9\%,$$

while the probability for a C_{f_2} candidate is

$$p(C_{f_2}) = \frac{1000}{1000 + 1000} = 50.0\%,$$

that is, a learned classifier will probably classify a C_{f_3} candidate as positive with higher confidence than a C_{f_2} candidate, indicating C_{f_3} is preferred over C_{f_2} for the antecedent. This is contradictive to the predefined preference relationships.

4.2.2 Resolution

The single-candidate model will also encounter problems in resolution. As in the model candidates are always considered in isolation, it is very likely that two or more candidates are classified as positive, i.e., coreferential to the anaphor. Therefore, some clustering strategy has to be applied to link an anaphor to a proper positive candidate. The *closest-first* strategy (Soon et al., 2001; Strube and Mueller, 2003), for example, links an anaphor to the closest candidate that is coreferential. The *best-first* strategy (Aone and Bennett, 1995; Ng and Cardie, 2003b; Yang et al., 2004a), on the other hand, links the anaphor to the positive candidate with the highest confidence value. And *aggressive-link* strategy (McCarthy and Lehnert, 1995) first merges all the chains of the coreferential NPs and then links the anaphor to the expanded chain.

Different clustering strategies lead to different resolution performance. Compared with *closest-first*, *best-first* is supposed to produce higher precision while *aggressive-link* ought to yield higher recall. However, all these strategies are made in an ad-hoc manner and are not necessarily the optimal from an empirical point of view (Ng, 2005).

4.3 The Twin-Candidate Model

Motivated by the above limitations of the single-candidate model, in this thesis we propose an alternative twin-candidate model to do antecedent selection. Different from the single-candidate model, the model explicitly learns a preference classifier to determine the preference relationships among the 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,

$$\begin{aligned} & p\ (\ ante(C_k) \mid ana, C_1, C_2, \ldots, C_n) \\ = \ & p\ (\ C_k \succ \{C_1, ..., C_{k-1}, C_{k+1}, ...C_n\} \mid ana, C_1, C_2, \ldots, C_n) \\ = \ & p(C_k \succ C_1, ..., C_k \succ C_{k-1}, C_k \succ C_{k+1}, ..., C_k \succ C_n \mid ana, C_1, C_2, \ldots, C_n) \end{aligned} \tag{4.4}$$

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

$$\begin{aligned} & p(C_k \succ C_1, ..., C_k \succ C_{k-1}, C_k \succ C_{k+1}, ..., C_k \succ C_n \mid ana, C_1, C_2, \ldots, C_n) \\ = \ & \prod_{1<i<n, i\neq k} p(C_k \succ C_i \mid ana, C_k, C_i) \end{aligned} \tag{4.5}$$

Thus,

$$\begin{aligned} & \ln p\ (\ ante(C_k) \mid ana, C_1, C_2, \ldots, C_n) \\ = \ & \sum_{1<i<n, i\neq k} \ln p(C_k \succ C_i \mid ana, C_k, C_i) \end{aligned} \tag{4.6}$$

$$\propto \sum_{1<i<n, i\neq k} \ln CF(\ \boldsymbol{i}\{ana, C_k, C_i\},\ C_k\ ) \tag{4.7}$$

This suggests that the possibility 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 will 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: $\boldsymbol{i}\{ana, C_i, C_j\}$, where *ana* is the possible 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, e.g., "01" indicating C_j is preferred over C_i while "10" indicating C_i is preferred. Trained on the instances built based on this principle, the classifier is capable of determining the preference between any two candidates of a given anaphor, by returning a class label, either "01" or "10" accordingly.

The key features of the twin-candidate model are:

1. Like the single-candidate model, the twin-candidate model has a limited class number and a fixed feature length (for the 3 elements in an instance). Therefore, it can avoid the problems of data sparseness and variable-length features from which the all-candidate model suffers.

2. In contrast to the single-candidate model, the preference between candidates can be explicitly captured in the twin-candidate model. The antecedent selection is based on the ranking of the candidates, which is more suitable for the nature of antecedent selection.

3. In contrast to the single-candidate model, the distribution of the classes in the twin-candidate model is much more balanced.

4. In contrast to the single-candidate model, the clustering of the new NP is more reasonable in that an anaphor is to be linked to the most preferred candidate, instead of those chosen by ad-hoc clustering strategies.

5. The model considers two candidates at a time, which makes it possible to apply any discriminative learning algorithm to learn the preference regularities. Moreover, that also makes it possible to use inter-candidate features to directly describe the relationships between two competing candidates.

4.4 Summary

In this chapter we discussed several possible learning models of the coreference resolution problem, including the all-candidate model, the single-candidate and the twin-candidate model. In the first part of the chapter, we analyzed the pros and cons of the single-candidate model that is widely used in coreference resolution systems. We demonstrated that compared with the all-candidate model, the single-candidate model has many advantages: limited and fixed class number and feature number. However, the model still has its limitations: it ignores the preference relationship between candidates. Motivated by this, we then proposed an alternative twin-candidate model which explicitly learns a preference classifier to determine the preference between candidates. The twin-candidate model is supposed to be able to overcome the problems with the single-candidate model.

In the next chapter, we will further present the construction of the twin-candidate model and its application to the coreference resolution task.

Chapter 5

The Twin-candidate Model and its Application for Coreference Resolution

The previous chapter gives a brief introduction to the basic idea of the twin-candidate model. However, the details of the twin-candidate model have not been disclosed. Another important problem that is not covered is how to apply the twin-candidate model to do coreference resolution. The basic twin-candidate model only focuses on antecedent selection. It always picks out a "best" candidate as the antecedent, even if the current NP to be resolved is not an anaphor. The model itself cannot identify and block the invalid resolution of non-anaphors. Therefore, some strategies have to be used to make it applicable to coreference resolution.

This chapter will have an in-depth study on the twin-candidate model. The first part of the chapter describes in detail the basic training and resolution procedures of the twin-candidate model, including instance construction, classifier generation, and antecedent determination. The second part explores strategies to deploy the

twin-candidate model to the coreference resolution task.

5.1 Structure of the Twin-candidate Model

As introduced in the previous chapter, the structure of the twin-candidate model is different from that of the single-candidate learning model in the following ways:

Instance: A training or testing instance is formed by an anaphor and a pair of antecedent candidates.

Classifier: The classifier is learned with the aim to explicitly determine the preference between two candidates to be antecedent.

Resolution: The antecedent is identified based on the ranking of the candidates, done by pairwise comparisons among the candidates.

5.1.1 Instance Representation

In the twin-candidate model, an instance takes a form of $\boldsymbol{i}\{Ana,\ C_i,\ C_j\}$, where Ana is a possible 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 labelled as "10" if C_i is preferred over C_j to be the antecedent, or "01" if otherwise.

An instance is associated with a feature vector. The features can describe the lexical, syntactic, semantic and positional relationships between Ana and each of the candidates, C_i or C_j. In addition, inter-candidate features can be used to represent the relationships between the pair of candidates, e.g. the distance in position between C_i and C_j. The features used in our study will be described in detail in the next chapter.

5.1.2 Training Instances Creation

In order to obtain a preference classifier, the training data should consist of the instances that have two candidates with an explicit preference relationship. As mentioned, given an anaphor to be resolved, the antecedent is the most preferred candidate among the candidate set. Thus a pair of candidates, one being coreferential and the other being non-coreferential, can be used safely to create a training instance. Note that a pair of non-coreferential candidates is not suitable for instance creation, because the preference between the pair, although it does exist, cannot be explicitly represented for training.

Based on this idea, for a given anaphor encountered during training, *Ana*, the closest coreferential candidate, C_{ante}, is used as the anchor candidate to compare against other non-coreferential candidates and create the training instances. An instance is labelled as "10" or "01" according to the positional relationship between the anchor candidate and the non-coreferential one, C_{nc}. Specifically, if C_{ante} is closer to *ana* than C_{nc}, the instance $\boldsymbol{i}\{Ana,\ C_{nc},\ C_{ante}\}$ is labelled as "01". Otherwise, if C_{nc} is closer, the instance $\boldsymbol{i}\{Ana,\ C_{ante},\ C_{nc}\}$ is labelled as "10" instead. (We will have a "00" label for another use, which will be explained later.)

Figure 5-1 shows the algorithm of the training instance generation.

Consider the following text, as an example:

[$_1$ 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] .

Table 5.1: An example text for instance creation in the twin-candidate model

In the above text segment, [$_4$ that company] and [$_5$ the money] are two anaphors with [$_1$ Globalstar] and [$_2$ $600 million] being their antecedents respectively. Thus the

Algorithm Training-Instance-Generation
Input:
DS: Training Document Set
Output:
IS: Training Instance Set

$IS = \emptyset$;
for each document d $\{NP_1, NP_2,\ldots,NP_n\}$ in DS
  for j = 1 **to** n
    if NP_j is not in any coreferential chain
      continue;

    $ante$ = the index of the immediate antecedent of NP_j;

    for i = j - 1 **downto** 1
      if NP_i is not a valid antecedent candidate
        continue;
      if NP_i is in the coreferential chain of NP_j
        continue;

      if $(i > ante)$
        $inst$ = Create_Inst(NP_j, NP_{ante}, NP_i);
        $IS = IS \cup \{<inst,$ "10"$>\}$;
      endif

      if $(i < ante)$
        $inst$ = Create_Inst(NP_j, NP_i, NP_{Ante});
        $IS = IS \cup \{<inst,$ "01"$>\}$;
      endif

    endfor

  endfor
endfor

return IS;

Figure 5-1: Training instance generation for the twin-candidate model

training instances to be created for this text are:

Instances	Label
i{[$_4$ that company], [$_1$ Globalstar], [$_2$ $600 million]}	10
i{[$_4$ that company], [$_1$ Globalstar], [$_3$ Schwartz]}	10
i{[$_5$ the money], [$_1$ Globalstar], [$_2$ $600 million]}	01
i{[$_5$ the money], [$_2$ $600 million], [$_3$ Schwartz]}	10
i{[$_5$ the money], [$_2$ $600 million], [$_4$ that company]}	10

5.1.3 Classifier Generation

Based on the set of feature vectors for the generated training instances, a classifier can be trained using a discriminative machine learning algorithm. Given the feature vector of a test instance ***i***{*Ana*, C_i, C_j} ($i < j$), the classifier is supposed to return a class label of "10" indicating that C_i is preferred over C_j for the antecedent of *Ana*; or "01" indicating that C_j is preferred.

5.1.4 Antecedent Identification

After being trained, the preference classifier can be used to select the antecedent for each anaphor encountered in a text. The antecedent identification procedure could be thought of as a tournament, a competition in which many participants play each other in individual matches. The candidates are just like the players in the 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 that judges which candidate wins or loses in a match.

Two competition schemes could be employed to pick out the antecedent from the given candidate set: Elimination and Round-Robin.

Elimination

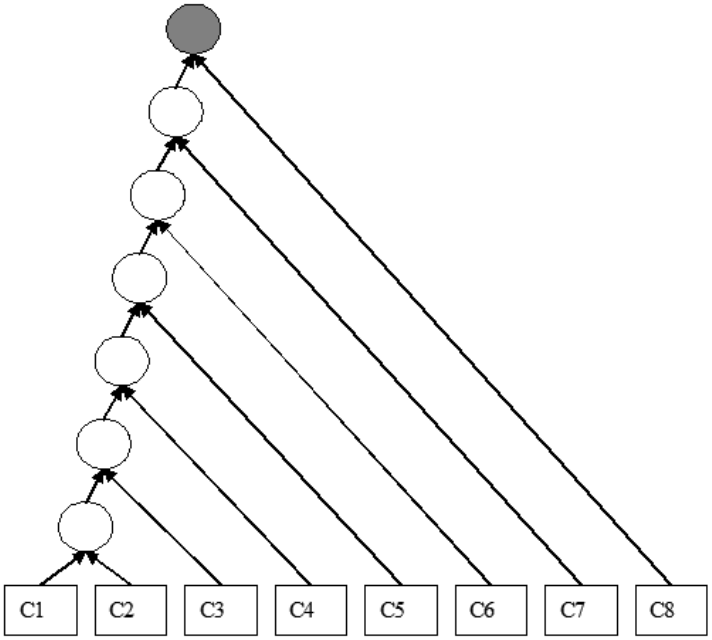
Elimination is a type of tournament where the loser in a match is eliminated. Such a scheme is also applicable to antecedent selection. Given an anaphor *Ana*, the candidates are paired one by one to create the test instances, $\boldsymbol{i}\{Ana, C_i, C_j\}$ $(i > j)$. A feature vector is associated with an instance and presented to the learned classifier, which then returns a class label, either "10" or "01" indicating which candidate is preferred over the other as the antecedent. The "losing" candidate, that is, the candidate that is less preferred, is eliminated and no longer considered in the subsequent comparisons. The antecedent is the candidate that wins the final comparison.

One simple elimination scheme, as adopted in Connolly et al. (1997)'s work (see Section 3.2.2), is linear elimination. In the scheme, the comparison starts from the first candidate and proceeds forwards in the positional order. The first candidate is compared with the second one. The less preferred candidate is eliminated and the winner is compared with the third candidate. The process continues until all the candidates are compared. The final comparison is made between the last candidate and the winner coming from the previous candidates. The top of Figure 5-2 illustrates the resolution of an anaphor with eight candidates using such a linear elimination scheme.

The problem with the linear scheme, however, is that it is unfair for the candidates that occur earlier. For example, the first candidate has to win all the opponents to be antecedent, whereas the last candidate only needs to win in one comparison. Although the order could be reversed so that the process starts with the last candidates and do comparisons backwards, the bias problem still exists.

A more reasonable procedure, as in the real-life tournament, is to compare the candidates in multiple rounds. In each round, a series of comparisons is done between consecutive candidates. The winners come into the next round while the losers are

Linear Elimination:



Multi-round Elimination:

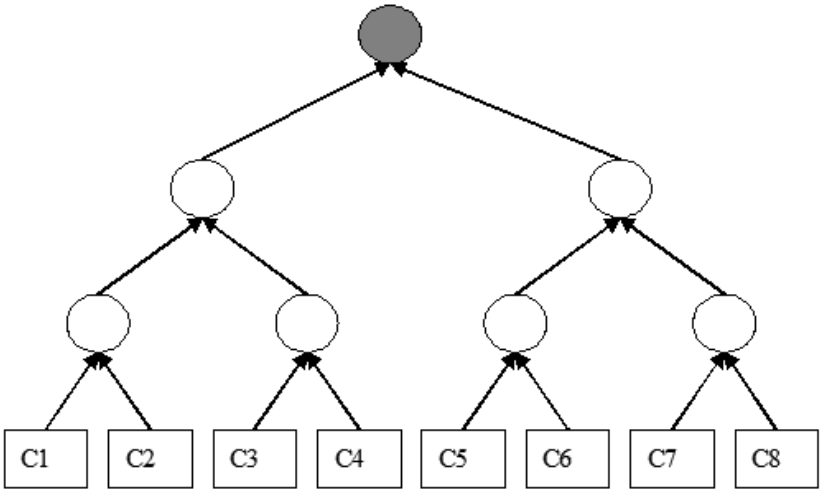


Figure 5-2: Illustration for antecedent selection using the elimination scheme

eliminated. The process continues until only one candidate remains, and this final winner will be selected as the antecedent. Such a multiple-round procedure can avoid the bias toward the latterly compared candidates. The bottom of Figure 5-2 shows the resolution procedure for an 8-candidate anaphor.

As an example to demonstrate the two elimination resolution schemes, consider the following text:

Any design to link China's accession to the WTO with [$_1$ the missile tests] was doomed to failure.

"If [$_2$ some countries] try to block China TO accession, that will not be popular and will fail to win the support of [$_3$ other countries]" she said.

Although [$_4$ no governments] have suggested [$_5$ formal sanctions] on China over [$_6$ the missile tests], the United States has called [$_7$ them] "provocative and reckless" and other countries said they could threaten Asian stability.

Table 5.2: An example text for antecedent selection

In the text there exists a coreferential chain: [$_1$ the missile tests] - [$_6$ the missile tests] - [$_7$ them]. Suppose we have a "perfect" classifier which can correctly determine the preference of the coreferential candidates over those non-coreferential ones (for two coreferential candidates, the one closer to the anaphor is preferred). We list the test instances to be generated for the resolution of [$_6$ the missile tests] and [$_7$ them], in Table 5.3 (for linear elimination) and Table 5.4 (for multi-round elimination).

The elimination scheme enables a relatively large number of candidates to be processed. Either the linear or the multi-round elimination has only O(N) computational complexity, where N is the number of the candidates. However, as in our twin-candidate model no constraints are imposed to enforce transitivity of the preference relation, the preference classifier would likely output $C_1 \succ C_2$, $C_2 > C_3$, and

Anaphor	Candidate1	Candidate2	Preference
[$_6$ the missile tests]	[$_1$ the missile tests]	[$_2$ some countries]	10
[$_6$ the missile tests]	[$_1$ the missile tests]	[$_3$ other countries]	10
[$_6$ the missile tests]	[$_1$ the missile tests]	[$_4$ no governments]	10
[$_6$ the missile tests]	[$_1$ the missile tests]	[$_5$ formal sanctions]	10
[$_7$ them]	[$_1$ the missile tests]	[$_2$ some countries]	10
[$_7$ them]	[$_1$ the missile tests]	[$_3$ other countries]	10
[$_7$ them]	[$_1$ the missile tests]	[$_4$ no governments]	10
[$_7$ them]	[$_1$ the missile tests]	[$_5$ formal sanctions]	10
[$_7$ them]	[$_1$ the missile tests]	[$_6$ the missile tests]	01

Table 5.3: The testing instances generated for the example text under the linear elimination resolution scheme

Anaphor	Candidate1	Candidate2	Round	Preference
[$_6$ the missile tests]	[$_1$ the missile tests]	[$_2$ some countries]	1	10
[$_6$ the missile tests]	[$_3$ other countries]	[$_4$ no governments]	1	01
[$_6$ the missile tests]	[$_1$ the missile tests]	[$_4$ no governments]	2	10
[$_6$ the missile tests]	[$_1$ the missile tests]	[$_5$ formal sanctions]	3	10
[$_7$ them]	[$_1$ the missile tests]	[$_2$ some countries]	1	10
[$_7$ them]	[$_3$ other countries]	[$_4$ no governments]	1	01
[$_7$ them]	[$_5$ formal sanctions]	[$_6$ the missile tests]	1	01
[$_7$ them]	[$_1$ the missile tests]	[$_4$ no governments]	2	10
[$_7$ them]	[$_1$ the missile tests]	[$_6$ the missile tests]	3	01

Table 5.4: The testing instances generated for the example text under the multi-round elimination resolution scheme

$C_3 \succ C_1$. Thus it is unreliable to eliminate a candidate once it happens to lose in one comparison, without considering all of its wining/losing results against the other candidates.

Round-Robin

In Section 4.3 we have shown that the probability that a candidate is the antecedent could 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

$$\begin{aligned} Antecedent(ana) &= \arg_i \max p\ (\ ante(C_i) \mid ana, C_1, C_2, \ldots, C_n) \\ &\propto \arg_i \max \sum_{j \neq i} CF(\boldsymbol{i}\{ana, C_i, C_j\}, C_i) \end{aligned} \tag{5.1}$$

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

$$Score(C_i) = \sum_{j \neq i} CF(\boldsymbol{i}\{ana, C_i, C_j\}, C_i) \tag{5.2}$$

then the best 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

$$Score(C_i) = |\{C_j | C_i \succ C_j\}| - |\{C_j | C_j \succ C_i\}| \tag{5.3}$$

That is, the score of a candidate is the difference between the number of the opponents to which it is preferred and the number of the opponents to which it is less preferred. To obtain the scores, the antecedent candidates of the current anaphor are compared with each other. The generated testing instances are presented to the

classifier to determine preference between any two candidates under consideration. For each candidate, its comparison result against every other candidate is recorded: its score increases by *one* if it is judged as preferred over a competing candidate in one comparison, or decreases by *one* if it is judged as less preferred.

Antecedent selection 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 picked out based on the winning-losing records of the participating players.

In contrast to the elimination scheme, the round-robin scheme is fair for each competitor as a losing candidate in a comparison is not knocked out instantly. Therefore, compared with the elimination scheme, this model is supposed to give more reliable ranking of the candidates.

The resolution algorithm for the antecedent selection using the round-robin scheme is shown in Figure 5-3. In the algorithm, the score of each candidate increases by one every time when it wins, or decreases by one when it loses. If two or more candidates have the same maximum score, the one closest to the anaphor is selected. The computational complexity for the algorithm to resolve an anaphor is $O(N^2)$, where N is the number of the candidates.

Consider the example in Table 5.1 again. Table 5.5 lists the test instance to be generated for resolving the anaphor [$_6$ the missile tests]. The scores of the candidates are summarized in Table 5.6. As observed, the candidate C_1 has the maximum score of 4, and thus it will be selected as the final antecedent.

For some machine learning algorithms, the learned classifier is able to classify a testing instance with a confidence value. We can use the confidence values, instead of the simple 0 and 1, to get a better estimation of the possibility of a candidate to be the antecedent, by summing up the confidence values returned by the classifier in judging the preference of the current candidate against the other ones. That is, the

Algorithm ANTE-SEL
Input:
ana: the anaphor under consideration
CS: the set of antecedent candidates of *ana*
output:
the index of the antecedent of *ana*

sort CS by the ascending order of position.

```
for i = 1 to |CS|
    Score[ i ] = 0;

for j = |CS| downto 2
    for i = j - 1 downto 1

        inst = create_inst(ana,CS[i],CS[j]);
        label = classify(inst);

        if (label == 01)
            Score[ i ]−−;
            Score[ j ]++;
        endif;

        if (label == 10)
            Score[ i ]++;
            Score[ j ]−−;
        endif;

    endfor;
 endfor;
```

$AnteIdx = \arg\max_i Score[i]$;
return $AnteIdx$

Figure 5-3: The antecedent selection algorithm using the round-robin resolution scheme

Anaphor	Candidate1	Candidate2	Preference
[$_6$ the missile tests]	[$_1$ the missile tests]	[$_2$ some countries]	10
[$_6$ the missile tests]	[$_1$ the missile tests]	[$_3$ other countries]	10
[$_6$ the missile tests]	[$_1$ the missile tests]	[$_4$ no governments]	10
[$_6$ the missile tests]	[$_1$ the missile tests]	[$_5$ formal sanctions]	10
[$_6$ the missile tests]	[$_2$ some countries]	[$_3$ other countries]	01
[$_6$ the missile tests]	[$_2$ some countries]	[$_4$ no governments]	01
[$_6$ the missile tests]	[$_2$ some countries]	[$_5$ formal sanctions]	01
[$_6$ the missile tests]	[$_3$ other countries]	[$_4$ no governments]	01
[$_6$ the missile tests]	[$_3$ other countries]	[$_5$ formal sanctions]	01
[$_6$ the missile tests]	[$_4$ no governments]	[$_5$ formal sanctions]	01

Table 5.5: The testing instances generated for the example text under the round-robin resolution scheme

	C1	C2	C3	C4	C5	Score
C1		+1	+1	+1	+1	4
C2	-1		-1	-1	-1	-4
C3	-1	+1		-1	-1	-2
C4	-1	+1	+1		-1	0
C5	-1	+1	+1	+1		2

Table 5.6: The scores generated for the example text under the round-robin resolution scheme

score of a candidate becomes:

$$Score(C_i) = \sum_{C_i \succ C_j} CF(C_i \succ C_j) - \sum_{C_j \succ C_i} CF(C_j \succ C_i) \tag{5.4}$$

Here CF is the confidence value returned by the preference classifier.

The algorithm of weight-based round-robin scheme is similar to that non-weight version listed in Figure 5-3, except that the classification confidence is used in place of "1" to increase or decrease the score of a candidate.

5.2 Deploying the Twin-Candidate Model for Coreference Resolution

In the previous section we described the basic twin-candidate model for antecedent selection for anaphors. In coreference resolution, however, not all encountered NPs are valid anaphors that have coreferential NPs to be found in the preceding text. Thus how to refrain from the resolution of non-anaphors becomes an issue. Generally, the single-candidate model can deal with the problem naturally as an encountered NP will not be resolved if all of its candidate are judged as negative. However, the twin-candidate model, which aims to identify the preference between candidates, always picks out a "best" candidate as the antecedent, even if the current NP is a non-anaphor. Therefore, to apply the basic twin-candidate model to coreference resolution, some additional effort has to be used. In this section, we would like to explore several strategies to deploy the model for the task of coreference resolution.

5.2.1 Using an Anaphoricity Determiner

A natural solution to directly deploy the twin-candidate model to coreference resolution is to use an anaphoricity determination (AD) module to identify the non-anaphoric NPs in advance.

Algorithm COREFERENCE-RESOLVE
Input:
NP: the NP list in the text to be resolved
Output:
the coreference chain in which Link[a] = Link[b] if a co-refers to b

```
for i = 1 to |NP|
    Link[ i ] = i;

for j = 1 to |NP|
    isAna = AnaDet(NP_j);
    if (isAna == 0)
       continue;
    idx = Ante_Sel(NP_j,{NP_1..NP_{j-1}});
    Link[ j ] = Link[ idx ];
endfor;

return Link
```

Figure 5-4: The coreference resolution algorithm by using an AD module

In this strategy, the training instance set is created and the preference classifier is trained in the same way as described in the previous section. However, in resolution, a separate anaphoricity determination module is first applied to an NP to determine whether or not it is a valid anaphor. If the current NP is judged as anaphoric by the AD module, then the antecedent selection algorithm is applied to find its antecedent as usual. Otherwise, the NP is just left unresolved.

The coreference resolution algorithm is listed in Figure 5-4.

In the algorithm, *Ante_Sel* could be any antecedent selection algorithm described in the last section. And *AnaDet* is the Anaphoricity Determination module that is expected to output a result of 1 or 0, indicating whether the input NP is an anaphor or not. The construction of AD module is completely independent of that of the coreference resolution module, and could be built using any technique, either learning based or heuristics based.

In fact, the practice of using an AD module can be also seen in the single-candidate based coreference resolution systems (Ng and Cardie, 2002a; Ng, 2004). This solution, however, heavily relies on the performance of the AD module. Both the recall and the precision of the AD module in determining the anaphors have considerable influence on the coreference resolution results. On the one hand, if the module recognizes a positive anaphor as negative, the NP will not be resolved and no candidate will be selected as the antecedent. On the other hand, if the module recognizes a negative anaphor as positive, the NP will be resolved to a false antecedent that in fact does not exist.

5.2.2 Using a Candidate Filter

The idea of using a candidate filter comes from the re-ranking technique that is popular in many NLP problems such as POS tagging, NP chunking, parsing and so on (e.g., Collins and Duffy (2002), Shen et al. (2003), Collins and Roark (2004), Charniak and Johnson (2005), among others). In re-ranking, a generative module is first used to produce a set of candidates. Then a discriminative classifier is applied to rank the candidates and select the best one as the final target.

Such an idea can be also applied to coreference resolution. At first, given the initial candidate set, we use a filter to generate a set of "qualified" candidates. And then, from the filtered candidates we use the twin-candidate module to select the best one as the antecedent. If no candidate remains after the filtering module, we consider the current NP non-anaphoric and leave it unresolved. Here the candidate filter serves two purposes: First, it acts as the generation module to provide qualified candidates for the twin-candidate model to do antecedent selection. Second, it acts as an anaphoricity determination module to block those non-anaphors, as described in the previous subsection.

Now the issue is: what is a "qualified" candidate? And how to filter those "unqualified" candidates?

Recall that we have compared the twin-candidate based antecedent selection as a tournament. The filtering step is like a qualification game taken before the tournament. Only those candidates good enough are qualified for the tournament. Motivated by this, in the first filtering step, each candidate is compared with a dummy "standard" candidate, denoted by C_0. If a candidate is preferred over C_0, it is eligible for the next round, otherwise it is eliminated instantly.

We can use a machine learning algorithm to build such a qualifier. In training, we stipulate that only those candidates coreferential to the anaphor are qualified and are preferred over C_0, while the others are unqualified and are less preferred over C_0. Based on this, we build a "10" training instance for each coreferential candidate C_a: $\boldsymbol{i}\{Ana, C_a, C_0\}$, and a "01" instance for each non-coreferential one C_{na}: $\boldsymbol{i}\{Ana, C_{na}, C_0\}$. Trained on such training instances, the classifier is able to judge whether a candidate is better than C_0 in resolution.

In fact, if we remove the dummy C_0 away in each training or testing instance, we can find that the filtering classifier is just a classifier trained under the single-candidate model. The filter built in this way actually uses the non-conditional probability that a candidate is coreferential to the anaphor as the threshold to remove candidates.

Based on this idea, we do coreference resolution as follows: for each NP encountered, a single-candidate based classifier is first applied to each of its antecedent candidates. The candidates judged as positive are kept while the candidates judged as negative are removed. If the remaining candidate set is empty, the current NP is regarded as non-anaphoric and left unresolved. Otherwise, the preference classifier is then applied to identify the best candidate, using the antecedent selection algorithm described in the last section. The coreference resolution algorithm is listed in Figure 5-5. In the algorithm, function *Filter* denotes the first-level filtering module,

Algorithm COREFERENCE-RESOLVE
Input:
NP: the NP list in the text to be resolved
Output:
the coreference chain in which Link[a] = Link[b] if a co-refers to b

```
for i = 1 to |NP|
    Link[ i ] = i;

for j = 1 to |NP|

    CS = ∅;
    for i = 1 to j - 1

        label = Filter(NP_j,NP_i)
        if ( label == 1 )
            CS = CS ∪ {NP_i};
        endif;

    endfor;

    if |CS| == 0
        continue;

    idx = Ante_Sel(NP_j,CS)
    Link[ j ] = Link[ idx ]

endfor;

return Link
```

Figure 5-5: The algorithm for coreference resolution by using a candidate filter

which returns a value of "0" or "1" indicating whether or not the current candidate should be removed from the candidate set.

The hybrid strategy combines the advantages of the single-candidate model and the twin-candidate model. One the one hand, as described, the first-level module acts as an anaphoricity determination to avoid the resolution of non-anaphors, and provides the second-level "qualified" candidates, which reduces the risk of selecting the wrong antecedent. On the other hand, the second-level module is capable of giving more accurate ranking of the candidates output by the first-level module.

However, similar to using the anaphoricity determiner, this strategy also has to rely on the performance of the additional filtering module. Particularly, it has a higher requirement on the recall of the filter than the precision: If the filtered candidate set does not contain any candidate that is coreferential to the anaphor, the second-level twin-candidate model is definitely unable to give the correct antecedent.

5.2.3 Using a Threshold

The above two strategies require an additional module to determine the anaphoricity of encountered NPs. In these strategies, the coreference resolution performance heavily depends on the results of the AD module. Could the twin-candidate model itself do anaphoricity determination and antecedent selection all together?

One possible solution is to set a threshold to avoid selecting a candidate that wins with low confidence. The assumption behind this strategy is that, since we only use anaphors to create the training instances, the learned classifier will give a low confidence value to a test instance formed by a non-anaphor. Based on the assumption, if a given NP has no candidate that at least wins a candidate with confidence high enough, we will consider the NP non-anaphoric and leave it unresolved.

Thus, we make a modification to the original antecedent selection algorithm:

Algorithm COREFERENCE-RESOLVE
Input:
NP: the NP list in the text to be resolved
Output:
the coreference chain in which Link[a] = Link[b] if a co-refers to b

```
for i = 1 to |NP|
    Link[ i ] = i;
for i = 1 to |NP|
    idx = Ante_Sel_THRESH(NP_i,{NP_1..NP_{i-1}})
    if (idx >= 1)
        Link[ i ] = Link[ idx ];
endfor
return Link;

function AnteSel_THRESH(Ana, CS, THRESH)

Score[1..|CS|] = 0;
for j = |CS| downto 2
    for i = j - 1 downto 1

        inst = Create_inst(Ana, CS[i], CS[j]);
        < label, Cf > = Classify(inst);
        if ( Cf < THRESH )
            continue;
        if (label == 01)
            Score[ j ] ++;
            Score[ i ] --;
        endif;
        if (label == 10)
            Score[ j ] --;
            Score[ i ] ++;
        endif;

    endfor;
endfor;

AnteIdx = arg max_i Score[i];
return AnteIdx
```

Figure 5-6: The algorithm for coreference resolution by using a threshold

- For the elimination scheme, we stipulate that the final winner should have to win at least one opponent with confidence above the specific threshold. If the maximum confidence value that the winner has ever obtained when compared with the opponents is less than the threshold, the winner is abandoned and the current NP is left unresolved.

- For the round-robin scheme, given two candidates under consideration, we update their match records only if the preference confidence value is above the specified threshold. If no candidate has a positive score in the end, we regard the NP in question as non-anaphoric and leave it unresolved[1]. In other words, an NP is resolved to a candidate only if the candidate wins against at least one competitor with confidence above the threshold. Figure 5-6 describes the resolution algorithm.

In the case when an NP has only one antecedent candidate, a pseudo-instance is created by pairing the candidate with itself. The NP is resolved to the candidate only if the confidence value is above the threshold.

As mentioned, the assumption behind this strategy is that the classifier returns low confidence for the test instances formed by non-anaphors. Although it may be true, there exist other cases in which the classifier also assigns low confidence values, for example, when the two candidates of an anaphoric NP both have strong preference as the antecedent. The solution of using a threshold cannot distinguish these different cases and thus may not be reliable for coreference resolution.

[1]Recall that we use a value from -1 to 1 to reflect the likelihood one candidate is preferred over the other. A final score of below zero indicates that the candidate is more likely not the antecedent than otherwise.

5.2.4 Using a Modified Twin-Candidate Model

In our study, we propose a modified twin-candidate model that can deal with the problem with the above three strategies. In the model, we try to teach the classifier to explicitly identify the cases of non-anaphors, instead of using an additional AD module or using a threshold implicitly. To do this, we provide a special set of instances formed by the non-anaphors to train the classifier. Given a test instance formed by a non-anaphor, the newly learned classifier is supposed to give a class label different from the instances formed by anaphors. This special label would indicate that the current NP is a non-anaphor, and no preference relationship is held between the two candidates under consideration. In this way, the twin-candidate model can do the anaphoricity determination by itself. We will describe the modified training and resolution procedures in this subsection.

Training

Like in the basic model, an instance in the modified twin-candidate model also takes a form of $\boldsymbol{i}\{Ana,\ C_i,\ C_j\}$. During training, for an encountered anaphor, we create "01" or "10" training instances in the same way as in the original learning framework, while for a non-anaphor *NonAna*, we do the following:

- From the candidate set, randomly select a candidate C_{rand} as the anchor candidate.
- Create a set of instances which is formed by *NonAna*, C_{rand} and each other non-coreferential candidate, C_{nc}.

The above instances formed by non-anaphors are labelled as "00". Note that an instance may have a form of $\boldsymbol{i}\{NonAna,\ C_{nc},\ C_{rand}\}$ if candidate C_{nc} is preceding C_{rand}, or like $\boldsymbol{i}\{NonAna,\ C_{rand},\ C_{nc}\}$ if candidate C_{nc} is following C_{rand}.

[$_1$ 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] .

Consider the text in Table 5.1, which is repeated here:

For the non-anaphors [$_3$ Schwartz] and [$_6$ the debt market], Suppose the selected anchor candidates are [$_1$ Globalstar] and [$_2$ $600 million], respectively. The "00" instances generated for the text are:

i{[$_3$ Schwartz], [$_1$ Globalstar], [$_2$ $600 million]}	:	00
i{[$_6$ the debt market], [$_1$ Globalstar], [$_2$ $600 million]}	:	00
i{[$_6$ the debt market], [$_2$ $600 million], [$_3$ Schwartz]}	:	00
i{[$_6$ the debt market], [$_2$ $600 million], [$_4$ that company]}	:	00
i{[$_6$ the debt market], [$_2$ $600 million], [$_5$ the money]}	:	00

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}, the newly learned classifier is supposed to return "01" (or "10") indicating *Ana* is an anaphor and C_i (or C_j) is preferred to be its antecedent, or return "00" indicating *Ana* is a non-anaphor and no preference exists between C_i and C_j.

Resolution

Accordingly, we make a modification to the original resolution procedure under both the elimination and the round-robin schemes:

Elimination Scheme. For the elimination scheme, consecutive candidates are compared with each other. If the instance for two competing candidates is classified as "01" or "10", the preferred candidate is compared with subsequent competitors and the loser is eliminated immediately as normal. If the instance is classified as "00", both of the two candidates are discarded and never considered. For the linear elimination

scheme, the comparison restarts with the pair of the next two candidates, while for the multi-round elimination, neither of the candidates come into the subsequent rounds. If finally no candidate wins out, the new NP is considered as non-anaphoric and left unresolved.

Round-robin Scheme: The resolution procedure for the round-robin scheme is described in Figure 5-7. Like in the original algorithm, each candidate is compared with every other candidate. The difference is that, if two candidates are labelled as "00" in a match, both candidates receive a penalty of -1 (or $-CF$ in the weighted scheme) in their respective scores; If no candidate has a positive final score, then the NP is considered as non-anaphoric and left unresolved. Otherwise, it is resolved to the candidate with the highest score as usual.

When an NP to be resolved has only one antecedent candidate, a pseudo-instance is created by pairing the candidate with itself. The NP will not be resolved to the candidate if the instance is labelled as "00".

In the algorithm we also use a threshold: the scores of two candidate are updated only if the confidence for their preference is high enough. Note that different from the algorithm described in Section 5.2.3, the purpose of a threshold in this algorithm is to optimize the performance of the resolution, but not to identify the non-anaphors. In Chapter 7 we will further evaluate the influence of the threshold in the two algorithms.

Coreference resolution using our modified twin-candidate model has several advantages over the previously described strategies. Compared with the strategies of using an anaphoricity determiner or using a candidate filter, it requires no additional model and thus avoids the reliance on the performance of the other modules. Compared with the solution of using a threshold, it employs a learned classifier to explicitly identify the instances formed by non-anaphors, which is more reliable than implicitly depending on the classification confidence.

Algorithm COREFERENCE-RESOLVE
Input:
NP: the NP list in the text to be resolved
Output:
the coreference chain in which Link[a] = Link[b] if a co-refers to b

```
for i = 1 to |NP|
    Link[ i ] = i;
for i = 1 to |NP|
    idx = Ante_Sel_NEW(NP_i,{NP_1..NP_{i-1}})
    if (idx >= 1)
        Link[ i ] = Link[ idx ];
endfor
return Link;

function AnteSel_NEW(Ana, CS, THRESH)
Score[1..|CS|] = 0;
for j = |CS| downto 2
    for i = j - 1 downto 1
        inst = Create_Inst(Ana, CS[i], C[j]);
        < label, Cf > = Classify(inst);
        if (Cf < THRESH)
            continue;
        if (label == 01)
            Score[ i ]--;
            Score[ j ]++;
        endif;
        if (label == 10)
            Score[ i ]++;
            Score[ j ]--;
        endif;
        if (label == 00)
            Score[ i ]--;
            Score[ j ]--;
        endif;
    endfor;
endfor;
AnteIdx = arg max_i Score[i];
return AnteIdx
```

Figure 5-7: The algorithm for coreference resolution using the modified twin-candidate model

5.3 Summary

In this chapter, we had an in-depth exploration on the twin-candidate model. We divided the whole chapter into two parts. In the first part, we described in great detail the construction of the twin-candidate model, including instance representation, training procedure and resolution procedure. Particularly, we discussed two resolution schemes, *elimination* and *round-robin*, to select the antecedent from a candidate set. While the former is efficient, the latter is more reliable to represent the ranking of the candidates.

In the second part, we investigated how to deploy the twin-candidate model into coreference resolution. The basic twin-candidate model does not handle non-anaphors in resolution. To address this problem, we proposed several feasible strategies. The first one uses an anaphoricity determination module to remove the non-anaphor before applying the twin-candidate model, while the second one uses a single-candidate classifier as a filter to output a set of candidates from which the twin-candidate classifier single out the best preferred one. The third one uses a threshold to block the invalid resolution of non-anaphors, with the assumption that the preference classifier gives a low confidence value to the test instances formed by non-anaphors. However, we showed that these strategies have their limitations. Finally, we proposed a modified twin-candidate model that can generate a preference classifier with a non-anaphor identification capability, which makes it possible to directly deploy the twin-candidate model for coreference resolution.

In the next chapter, we will discuss the knowledge representation problem for the twin-candidate model.

Chapter 6

Knowledge Representation for the Twin-Candidate Model

The previous chapter has given a detailed description of the twin-candidate model and its application to the coreference resolution task. However, the knowledge representation problem, which is a key issue for a learning based approach, has yet to be discussed. For example, how should one incorporate and organize the different sources of knowledge into the twin-candidate model? What kinds of knowledge should be used to indicate the preference relationship between candidates? And how does one obtain such knowledge?

This chapter will explore the knowledge representation problem for the twin-candidate model. The first part introduces a way to represent the knowledge, in terms of features, for preference determination. The second part discusses the feature selection and categorization. It gives a detailed description of each of the features used in our study, including their definition and computation.

6.1 Knowledge Organization

As mentioned, in the single-candidate model, the purpose of classification is to determine the coreference relationship between an anaphor and one individual antecedent candidate. Therefore, the knowledge for resolution is restricted to the anaphor and the candidate only. Specifically, given a 2-tuple instance $\boldsymbol{i}\{Ana\ ,\ C_i\}$, the knowledge can be categorized into the following three groups:

- Knowledge related to the possible anaphor, Ana
- Knowledge related to the individual candidate, C_i
- Knowledge related to the relationships between the candidate (C_i) and the anaphor (Ana).

In contrast, classification in the twin-candidate model is to determine the preference relationship between two competing candidates of a given anaphor. Thus we make use of two types of knowledge: The first type is the knowledge related to the individual candidates, C_i or C_j and their respective relationships with Ana. We assume the learning algorithm is able to compare the properties of the two candidates and then find the preference regularities for learning and testing. The second type is the knowledge that directly captures the preference between the two candidates, e.g. "which candidate has a higher string similarity with anaphor than the other?" This type of knowledge can explicitly represent the preference factors between candidates, instead of depending on the learning algorithm to discover them.

Specifically, given a 3-tuple instance $\boldsymbol{i}\{Ana,\ C_i,\ C_j\}$, the knowledge can be categorized into six groups as follows:

- Knowledge related to the possible anaphor, Ana
- Knowledge related to the individual candidate, C_i

- Knowledge related to the relationships between the candidate (C_i) and the anaphor (Ana)
- Knowledge related to the individual candidate, C_j
- Knowledge related to the relationships between the candidate (C_j) and the anaphor (Ana)
- Knowledge related to the relationships between the two competing candidates (C_i and C_j)

6.2 Features Definition

Knowledge is usually represented in terms of *feature* in supervised learning. The selection and definition of features have a significant influence on the performance of a learning based system. For the coreference resolution task, what kinds of features should be used to achieve a good performance still remains an open problem[1]. One issue that is often in argument is whether we should use domain-specific or domain-independent features.

Domain-specific features can lead to effective resolution in a particular domain (McCarthy, 1996). Nevertheless, they lack adaptivity: features effective in one domain may not necessarily work equally well when applied to other domains. By contrast, domain-independent features are suitable for various domains and, as revealed by recent research on coreference resolution, can also bring encouraging results (Mitkov, 1998; Soon et al., 2001). In our study, the features used are similar to those in (Soon et al., 2001) (See Section 3.2.2), which are restricted to the domain-independent ones. All the features can be easily obtained from the output of pre-processing modules or

[1]Ng and Cardie (2002b) examine a large number of features that include a variety of linguistic constraints and preferences, and find the performance on this full set of feature performs significantly worse than on a smaller set manually fine-tuned feature set.

other reliable resources. Their utility for coreference resolution has been proven in many previous research works.

In this section we will introduce the feature set for the twin-candidate model adopted in our study. Throughout the section, *Ana* refers to the possible anaphor to be resolved, while C_i and C_j refer to the two competing candidates under consideration.

6.2.1 Features Related to the Anaphor

ana_Def

Is the anaphor a definite noun phrase?
The possible values are 0, 1.

Definite noun phrases include definite descriptions and demonstrative descriptions. The former are NPs that start with the word *the*, for example, "the company", while the later are those that start with demonstrative determiners like *this*, *that*, *these*, or *those*.

Most NPs that start with a definite article or a demonstrative are anaphoric and should be resolved to one previous NP[2]. If the possible anaphor *Ana* is a definite NP, return 1; else return 0.

ana_InDef

Is the anaphor an indefinite noun phrase?
The possible values are 0, 1.

An indefinite noun phrase starts with the article *a* or *an*, for example, "a company"

[2]A definite description may also be used as non-anaphoric in a large situation such as "the moon", "She jumps at the slightest noise." "Great changes have taken place in the place where he lived.". See (Poesio and Vieira, 1998) for a deeper exploration on the use of the definite description in reference resolution.

or "an orange". Indefinite NPs usually introduce new entities into a text, thus should not be resolved to any previously mentioned NP[3]. If *Ana* is an indefinite noun, return 1; else return 0.

ana_Name

Is the anaphor a proper name?
The possible values are 0, 1.

In our study a proper name, like "Microsoft" or "Bill Gates", is determined based the results of a named-entity recognition (NER) module (The NER module used in our system will be described in Chapter 7.). If *Ana* is a proper name, return 1; else return 0.

ana_Pron

Is the anaphor a pronoun?
The possible values are 0, 1.

A pronoun usually refers to a noun phrase previously mentioned[4]. If the anaphor is a pronoun, return 1; else return 0. As the property of a pronominal anaphor may play a role in its antecedent selection, we use another two additional features to further describe pronouns.

ana_Reflexive

Is the anaphor a reflexive pronoun?
The possible values are 0, 1.

[3]An indefinite NP, when used as the appositional phrase or a predicate nominal, could possibly be coreferential to the previous NP. For example, "Julius Caesar, a well-known emperor,...", or "Mediation is a viable alterative to bankruptcy".

[4]In the case of cataphora or in pleonastic use, a pronoun may have no antecedent in the previous text, for example, "Because she has grown up, Kate was asked to do this task alone" and "it is raining".

If *Ana* is a reflexive pronouns (like "himself", "themselves"), return 1; else return 0.

ana_PronType

What is the type of the anaphor if it is a pronoun?
The possible values are 0, 1, 2, 3, 4.

We distinguish between four types of pronouns:

1. First or second person pronoun, like "we", "I", "you", ...
2. Single third-person pronoun that refers to human beings, like "he", "she", ...
3. Single third-person that refers to non-human beings, like "it",...
4. Plural third-person pronoun, like "they", ...

If *Ana* is not a pronoun, return 0. Otherwise return $1 \sim 4$ according to the category to which it belongs.

6.2.2 Features Related to the Individual Candidate

candi_Def

Is the candidate a definite noun phrase?
The possible values are 0, 1.

A definite noun phrase is often anaphoric and thus is a *hearer-old discourse entity* that, as proposed by Strube (1998), should be preferred over other *mediate* or *hearer-new* ones for the antecedent in the candidate ranking. If C_i (C_j) is a definite description, return 1; else return 0.

candi_InDef

Is the candidate an indefinite noun phrase?
The possible values are 0, 1.

If C_i (C_j) is an indefinite NP that starts with "a" or "an", return 1; else return 0.

candi_Name

Is the candidate a proper name?
The possible values are 0, 1.

As described in Strube (1998), a proper name is also a *hearer-old* discourse entity (either "evoked" or "unused"), and therefore may have higher preference as the antecedent. If C_i (C_j) is a proper name, return 1; else return 0.

candi_Pron

Is the candidate a pronoun?
The possible values are 0, 1.

Like a definite candidate, a pronominal candidate is an "evoked" *hearer-old* discourse entity and should rank higher than other antecedent candidates (Strube, 1998). If C_i (C_j) is a pronoun, return 1; else return 0.

candi_FirstNP

Is the candidate the first occurring NP in its current sentence?
The possible values are 0, 1.

A candidate that is the first NP in a sentence is a salience indicator of the candidate (Mitkov, 1998). In the following sentence:

(**Eg 6.1**) "**IBM**'s accounting grew much more liberal since the mid 1980s as **its** business turned sours".

the antecedent candidate "IBM" is the first NP in the sentence. Thus for "IBM", the feature *candi_FirstNP* returns 1; while for the other candidates in this sentence, the feature returns 0.

6.2.3 Features Related to the Candidate and the Anaphor

SameSent

Do the candidate and the anaphor occur in the same sentence?
The possible values are 0, 1.

The feature captures the distance relationship between the candidate and the anaphor. If the candidate and the anaphor are in the same sentence, return 1. If they are one or more sentences apart, return 0.

NearestNP

Is the candidate the preceding NP closest to the anaphor?
The possible values are 0, 1.

This feature represents another distance relationship between the candidate and the anaphor. If a candidate is the one closest to the anaphor, return 1, else return 0.

NameAlias

Are the candidate and the anaphor the name alias of each other?
The possible values are: 0, 1.

A name occurring in a text could be subsequently mentioned in a shortened version, or alias. Two NPs that are alias of each other very likely refer to the same entity

and thus are probably considered coreferential. In our study, alias is determined in different ways according to the type of names. For temporal names, like "04/07" and Jul 4th", the day, month and year values are to be extracted and compared. For personal names, e.g., "Consuela Washington" and "Ms. Washington", their last names are compared. For other types of names, a straightforward acronym examination is done by checking the first capitalized letter in each word of the name, for example, "SEC" and "the Securities and Exchange Commission". If the candidate C_j (C_j) and *Ana* are both names and are judged as alias of each other, return 1; else return 0.

Appositive

Are the candidate and the anaphor in the same appositive structure?
The possible value are: 0, 1.

Appositive is another important factor for coreference determination. Typically, an appositive structure is to provide an alternative description or name for an entity. Two NPs in an appositive structure are thus probably coreferential to each other[5]. For example, in the sentence:

(**Eg 6.2**) "Bill Gates, the CEO of Microsoft, ..."

Here "the CEO of Microsoft" and "Bill Gates" are in an appositive structure and are a coreferential pair.

In our study a set of heuristics is used to identify the appositive structure, for example, by checking the existence of verbs and punctuation marks around the NPs. If the candidate is judged to be appositive to the anaphor, return 1; else return 0.

[5]Different annotation guidelines may have different requirements on the appositional phrase. For example, in MUC-6, the coreference relation is marked only if the appositional phrase is a definite NP, while in MUC-7 both indefinite and definite noun phrase are possible.

NumberAgree

Do the candidate and the anaphor agree in number?

The possible values are 0, 1.

Number agreement is often used as a hard constraint to preclude two non-coreferential NPs. The computation of the feature is done by examining whether or not the two NPs are both singular and both plural. For pronouns, "they" are plural, while "it", "she", "he" and so on, are singular. For non-pronouns, the morphological head word is checked to determine whether is it is singular or plural. Some noun phrases may have a singular form but refer to a collective entity, such as "people", "police", "family" and so on. In our study we keep a list of such words to check whether an NP is in the list and should be considered as plural.

If the candidate and the anaphor do not have the same number agreement, return 0, else return 1.

GenderAgree

Do the candidate and the anaphor agree in gender?

The possible values are 0, 1.

Similar to number agreement, gender agreement is another commonly used constraint factor. A male person is definitely not coreferential to a female one. In our study, the gender of a noun phrase is determined in several ways. For a personal pronoun, "he" refers to a man while "she" refers to a woman[6]. For a personal name, the designator could indicate the gender, for example "Mr.", "Mrs", "Miss" and so on. If a name has no designator before it, we check through the whole document to see whether a later mention has one. For example, given "Consuela Washington... Ms. Washington", we can know the first name refers to a woman according to the desig-

[6]In some cases, "she" could refer to an inanimate thing like "ship" or "nation".

nator of the second mention of the name. Finally, for other general noun phrases, we use WordNet (Miller, 1990) to check whether the head of an NP has senses containing the key words like "woman", "girl", "female" and so on. If the candidate and the anaphor do not explicitly violate the gender agreement, return 1, else return 0.

HeadStrMatch

Do the candidate and the anaphor have the same head word?
The possible values are 0 or 1.

String matching has been recognized as a very important factor for coreference resolution in many research works (Strube et al., 2002; Yang et al., 2004c). In our study we use several features to represent this factor. The first one is for head-string matching.

The same head word provides a clue that two noun phrases may possibly refer to the same entity, for example, "A company,...,The big company...". Nevertheless, in many cases two NPs with the same head words do not really co-refer, for example, "the small company", "the big company".

If the candidate and the anaphor have the same head word, return 1; else return 0.

FullStrMatch

Do the candidate and the anaphor consist of the same strings?
The possible values are 0 or 1.

In contrast to head-string matching, full-string matching have strict examination on the strings contained in two noun phrases. It requires that the two NPs not only have the same head word, but also have the same modifiers (except for the articles and determiners like "this" and "that"). For example, "a company" and "the big

company" are not full-string matched, but "a big company" and "the big company" are. If the candidate and the anaphor match in full-string, return 1; else return 0.

Compared to *HeadStrMatch*, *FullStrMatch* identifies coreferential pairs with high precision, but it would nevertheless miss many positive cases[7].

StrSim

To what degree do the candidate and the anaphor match in strings.
The possible values range from 0 to 100.

As above mentioned, the simple head-string or full-string checking is not sufficient for coreference resolution. Standing between the two extremes, the feature *StrSim* measures to what degree two NPs match in strings. We use the following metric to evaluate the matching degree:

$$\text{CommonRatio(NP1, NP2)} = 100 \times \frac{|Str_{NP1} \cap Str_{NP2}|}{|Str_{NP1}|} \tag{6.1}$$

in which STR_{NP} is the string list of the words contained in NP (excluding the articles and determiners). We define the feature *StrSim* as follows:

$$StrSim = \begin{cases} \text{CommonRatio}(Ana,\ C) & : \quad Ana \text{ and } C \text{ have the same head} \\ 0 & : \quad otherwise \end{cases} \tag{6.2}$$

For example, if the anaphor is "the company" and the candidate is "the big company", the feature returns 100. If the anaphor is "'the big company" and the candidate is "the company", the feature returns 50.

SemSim

The semantic compatibility between the non-pronominal anaphor/candidate pair.
The possible values range from 0 to 100.

[7]Soon et al. (2001) reported that both head-string matching and full-string matching alone lead to similar F-measure, with trade-offs between recall and precision, on both MUC-6 and MUC-7 data set.

Two NPs that refer to the same entity must have the compatible semantic category. We use WordNet distance (Poesio et al., 2004) to measure the semantic compatibility between non-pronominal NPs, which is calculated as follows:

1. Obtain from WordNet all the senses of the candidate and anaphor, $\{S_{C_i}\}$ and $\{S_{Ana_i}\}$[8].

2. Get the hypernym tree of each of the senses.

3. For each sense pair, S_{C_i} and S_{Ana_i}, find the most specific common subsumer S_{ij}^{Comm}, i.e., the closest concept which is the hypernym of S_{C_i} and S_{Ana_i}

4. Let distance(S, S') be the number of hypernym links between concept S and S'. The shortest distance between the candidate and anaphor is computed as ShortWNDist(C, Ana)

$$= \min_{i,j} \min\{distance(S_{C_i}, S_{ij}^{Comm}), distance(S_{Ana_j}, S_{ij}^{Comm})\} \tag{6.3}$$

5. The normalized WordNet distance, WNDist, is obtained by dividing the shortest distance by a MaxWNDist factor (15 in our study).

$$\text{WNDist}(C, Ana) = \begin{cases} 1 & : \text{ no common subsumer} \\ \frac{\text{ShortWNDist}_{(C,Ana)}}{\text{MaxWNDist}} & : otherwise \end{cases} \tag{6.4}$$

6. The value of *SemSim* is simply 100*(1 - WNDist(C, Ana))

SemSimPron

The semantic compatibility between the pronominal anaphor and the candidate.
The possible values range from 0 $\sim +\infty$.

[8]If a noun phrase is a name, we can think of the NP as a dummy word that has one sense returned from the named-entity recognition module.

Since pronouns, especially neutral pronouns, carry little semantics of their own, the semantic compatibility between a pronominal anaphor and its antecedent candidate is commonly evaluated by examining the relationships between the candidate and the anaphor's context, based on the statistics that the corresponding predicate-argument tuples occur in a particular large corpus. Consider the example given in the work of Dagan and Itai (1990):

(Eg 6.3) *They know full well that companies held tax money aside for collection later on the basis that the government said* ***it***$_1$ *was going to collect* ***it***$_2$.

For anaphor it_1, the candidate *government* should have higher semantic compatibility than *money* because *government_collect* is supposed to occur more frequently than *money_collect* in a large corpus. A similar pattern can also be observed for it_2.

Corpus-based semantic knowledge has been employed in several previous anaphora resolution work (Dagan and Itai, 1990; Bean and Riloff, 2004; Kehler et al., 2004). However, corpus-based statistics usually suffers from data-sparseness problems. That is, many predicate-argument tuples would be unseen even in a large corpus. A possible solution is the web. It is believed that the size of the web is thousands of times larger than normal large corpora, and the counts obtained from the web are highly correlated with the counts from large balanced corpora for predicate-argument bi-grams (Keller and Lapata, 2003). So far the web has been utilized in nominal anaphora resolution (Modjeska et al., 2003; Poesio et al., 2004) to determine the semantic relation between an anaphor and candidate pair. In our approach, we also use the web to obtain the semantic compatibility for pronominal anaphors[9].

Three types of predicate-argument relationships, subject-verb, verb-object and possessive-noun, are considered in our work. Queries are constructed in the form of "NP$_{candi}$ VP" (for subject-verb), "VP NP$_{candi}$" (for verb-object), and "NP$_{candi}$'s NP"

[9]Detailed description of this feature is given in our work in (Yang et al., 2005b).

or "NP of NP_{candi}" (for possessive-noun). Consider the following sentence:

(**Eg 6.4**) *Several experts suggested that IBM's accounting grew much more liberal since the mid 1980s as* ***its*** *business turned sour.*

For the pronoun "*its*" and the candidate "*IBM*", the two generated queries are "*business of IBM*" and "*IBM's business*".

To avoid data sparseness, in an initial query only the nominal or verbal heads are retained. Also, each NE is replaced by the corresponding common noun. For example, "*IBM's business*" becomes "*company's business*".

A set of inflected queries is generated by expanding a term into all its possible morphological forms. For example, "*collect money*" becomes "*collected|collecting|... money*", and "*business of company*" becomes "*business of the company|companies*". Determiners are inserted for all the nouns in a query. Specifically, if a noun is the candidate under consideration, only the definite article *the* is inserted. Otherwise, *a/an*, *the* and the empty determiners (for bare plurals) are added (e.g., "*the|a business of the company|companies*").

Queries are submitted to a particular web search engine (Google in our study). All queries are performed as exact matching. The semantic compatibility feature, *SemSimPron*, can be represented as:

$$\text{SemSimPron}(candi, ana) = count(candi, ana) \tag{6.5}$$

where $count(candi, ana)$ is the hit number of the inflected queries returned by the search engine[10].

[10]Normalization can be done on this feature, which however made no much significant difference in the resolution performance as reported in the previous work (Yang et al., 2005b).

6.2.4 Features Related to the Competing Candidates

For some features like semantic or string similarity, the value may indicate the preference of a candidate to be the antecedent, i.e., the higher the value, the more preferred the candidate. The twin-candidate model makes it possible to consider the feature values of two candidates at a time. However, the learning algorithm is not necessarily powerful enough to discover the preference pattern (e.g. $SemSim_{Ci} > SemSim_{C_j}$ indicates $C_i \succ C_j$). Therefore, in our study we use a set of inter-candidate features to explicitly record the comparisons between the values of these features. These features can make the preference relationship between two candidates clearer, and thus are beneficial for both preference learning and preference determination.

inter_SameSent

Do the two candidates occur in the same sentence?
The possible values are 0, 1.

Feature *SameSent* records whether the candidate and the anaphor are in the same sentence. *inter_SameSent* further represents the positional relationship between the two competing candidates. If C_i and C_j are in the same sentence, return 1; if they are one or more sentences apart, return 0. For example,

(**Eg 6.5**) *"If [$_2$ some countries] try to block China TO accession, that will not be popular and will fail to win the support of [$_3$ other countries]" she said.*
Although [$_4$ no governments] have suggested [$_5$ formal sanctions] on China over [$_6$ the missile tests], the United States has called [$_7$ them]

For the instance ***i***{[$_7$ them], [$_5$ formal sanctions], [$_6$ the missile tests]}, the feature returns 0 because [$_5$ formal sanctions] and [$_6$ the missile tests] are in the

same sentence, while for ***i***{[$_7$ them], [$_2$ some countries], [$_6$ the missile tests]}, the feature returns 1.

inter_BetterStrSim

Which candidate has a higher string matching similarity with the anaphor?
The possible values are 0, 1, 2.

inter_BetterStrSim compares the feature *StrSim* of C_i and C_j. If C_i has a higher feature value than C_j, return 1. If C_j has a higher value, return 2. Otherwise if they have an equal value, return 0;

For example, in the sentence:

(**Eg 6.6**) *[$_1$ Jenny] bought [$_2$ the nice cup] [$_3$ last week]. However, [$_4$ yesterday] [$_5$ she] put [$_6$ the cup] on a plate and broke it.*

For the anaphor [$_6$ the cup], the feature *StrSim* of the candidate [$_2$ the nice cup] is 100, while those for [$_1$ Jenny] and [$_3$ last week] are 0. Thus the values of *inter_BetterStrSim* for the following instances are:
i{[$_6$ the cup], [$_1$ Jenny], [$_2$ the nice cup]} : 2
i{[$_6$ the cup], [$_2$ the nice cup], [$_3$ last week]} : 1
i{[$_6$ the cup], [$_1$ Jenny], [$_3$ last week]} : 0

inter_BetterSemSim

Which candidate has a higher semantic similarity with the anaphor?
The possible values are 0, 1, 2.

Similar to *inter_BetterStrSim*, this feature compares the feature *SemSim* of C_i and C_j. If C_i has a higher similarity value than C_j, return 1. If C_j has a higher one,

return 2; otherwise return 0.

inter_SemMagPron

The difference between the SemSim_Pron values of the candidates.
The possible values range from $-\infty \sim +\infty$.

Features *inter_BetterStrSim* and *inter_BetterSemSim* use 0, 1, 2 to represent the comparison of the corresponding feature values of C_i and C_j. As the semantic compatibility value for pronouns is a web count, we would like to use a feature to measure the degree that the value of *SemSimPron* for C_i is larger or smaller than that for C_j. Suppose the magnitude metric is defined as follows

$$mag(C_i, C_j) = \frac{SemSimPron_{C_i} + 1}{SemSimPron_{C_j} + 1}$$

we have the new feature:

$$inter\_SemMagPron(ana, C_i, C_j) = \begin{cases} mag - 1 & : \ mag >= 1 \\ 1 - mag^{-1} & : \ mag < 1 \end{cases}$$

The positive or negative value marks the times that $SemSimPron_{C_i}$ is larger or smaller than $SemSimPron_{C_i}$. Reconsider the sentence in Eg 6.3, which is repeated as follows:

They know full well that companies held [$_1$ tax money] aside for [$_2$ collection] later on [$_3$ the basis] that [$_4$ the government] said [$_5$ it] was going to collect [$_6$ it]

Suppose that for the anaphor [$_6$ it], the candidates [$_1$ tax money], [$_3$ the basis] and [$_4$ the government] have *SemSimPron* of 191000, 177 and 738 respectively. Thus for the instance $\boldsymbol{i}${[$_6$ it], [$_1$ Tax money], [$_4$ the government]}, *inter_SemMagPron* returns (191000+1)/(738+1) - 1 = 257,

and for the instance ***i***{[$_6$ it], [$_3$ basis], [$_4$ the government]}, it returns
1 - ((177+1)/(738+1))$^{-1}$ = -3.

6.3 Summary

In this chapter we discussed the knowledge representation problem of the twin-candidate model. In our study, two types of knowledge are used to represent the preference between candidates of a given anaphor. The first type is the knowledge related to the two individual candidates, while the second type is the knowledge that reflects the relationships between the two candidate, which can explicitly represent the preference factor for better learning and testing.

The definition of the features is vital for a learning-based system. In this chapter we gave a detailed description of the feature used in our study, which includes only those that are domain-independent and can be obtained easily from preprocessing modules or other reliable resources. The feature set can be divided into several categories: those related to the single candidate, those related to the anaphor, those related to the relationships between the candidate and the anaphor, and those related to the relationships between the two competing candidates. Especially, the last group of features is exclusive for the twin-candidate model to directly describe the preference factor between candidates. The whole feature set is summarized in Table 6.1.

The next chapter will have an in-depth evaluation of the twin-candidate model. The importance of different features in antecedent selection and coreference resolution will be analyzed in great detail.

Features describing the anaphor (*ana*):	
ana_Def	1 if *ana* is a definite noun phrase; else 0;
ana_Indef	1 if ana is an indefinite NP; else 0;
ana_Name	1 if *ana* is a proper name; else 0;
ana_Pron	1 if *ana* is a pronoun; else 0;
ana_Reflexive	1 if *ana* is a reflexive pronoun; else 0;
ana_PronType	Type of the *ana* if it is a pronoun;
Features describing the candidate (C_i and C_j)	
candi_Def	1 if C_i (C_j) is a definite noun phrase; else 0;
candi_Indef	1 if C_i (C_j) is an indefinite NP; else 0;
candi_Name	1 if C_i (C_j) is a proper name; else 0;
candi_Pron	1 if C_i (C_j) is a pronoun; else 0;
candi_FirstNP	1 if C_i (C_j) is the first NP in the sentence; else 0;
Features describing the relationships between C_i (C_j) and *ana*:	
SameSent	1 if C_i (C_j) and *ana* are in the same sentence; else 0;
NearestNP	1 if C_i (C_j) is the candidate closest to *ana*; else 0;
NameAlias	1 if C_i (C_j) and *ana* are in an alias of the other; else 0;
Appositive	1 if C_i (C_j) and *ana* are in an appositive structure; else 0;
NumberAgree	0 if C_i (C_j) and *ana* mismatch in the number agreement; else 1;
GenderAgree	0 if C_i (C_j) and *ana* mismatch in the gender agreement; else 1;
HeadStrMatch	1 if C_i (C_j) and *ana* have the same head string; else 0;
FullStrMatch	1 if C_i (C_j) and *ana* contain the same strings; else 0;
StrSim	The string similarity between C_i (C_j) and *ana*;
SemSim	The semantic similarity between the non-pronominal pair of C_i (C_j) and *ana*;
SemSimPron	The semantic similarity between C_i (C_j) and the pronominal *ana*;
Features describing the relationship between the two candidates	
inter_SameSent	1 if C_i and C_j are in the same sentence; else 0;
inter_BetterStrSim	Which candidate has a higher string similarity with *ana*;
inter_BetterSemSim	Which candidate has a higher semantic similarity with *ana*;
inter_SemMagPron	Magnitude between the *SemSimPron* of C_i and C_j;

Table 6.1: Feature set for coreference resolution using the twin-candidate model

Chapter 7

Evaluation

The previous chapters have described the twin-candidate model and its application to coreference resolution. But how does such a model work in real resolution? Is the model more effective than the single-candidate model in identifying the correct antecedent? What is the difference between the elimination and the round-robin resolution schemes? What impact do factors such as training size have on the resolution of the twin-candidate model as compared with the single-candidate model? And further, how does the twin-candidate model work for coreference resolution where the anaphoricity of an encountered NP is unknown? Is our modified twin-candidate model more effective than the single-candidate model, and than the twin-candidate model but with other resolution strategies?

This chapter will give an empirical evaluation of the twin-candidate model. The first part of the chapter describes the learning framework of the coreference resolution system. Then the rest of the chapter examines the efficacy of the twin-candidate model in antecedent identification for anaphora resolution, and further for coreference resolution.

7.1 Building a Coreference Resolution System

We built a coreference resolution system for evaluation. The whole system consists of two parts: training and resolution.

During training, an input annotated document is first processed by a pipeline of NLP modules to obtain all possible noun phrases as well as the necessary information for resolution. Training instances are then created in form of feature vectors, on the basis of the twin-candidate model. Given the feature vectors generated for the training documents, a classifier is generated using a certain machine learning algorithm.

During resolution, an input raw document is processed by the same NLP pre-processing modules as during training. For each NP encountered, a set of testing instances, also in form of feature vectors, is created for the possible antecedent candidates. The learned classifier is then applied to identify the antecedent, if any, to which the NP should be resolved. All the coreferential NPs are linked together as the output of the system.

Figure 7-1 illustrates the flowchart of the training and resolution procedures of the system. In the following subsections, we will give a brief introduction to the data set, the pre-processing NLP modules and the learning algorithm used in our system.

7.1.1 Corpus

Our system is run on MUC-6 and MUC-7 coreference corpora, the common data sets on which many coreference resolution systems are developed and evaluated. The documents in the data sets are the newswire articles from the Wall Street Journal, on the topic of business management.

As in the MUC-6 and MUC-7 coreference resolution task definition (Hirschman, 1998), the coreference relationship can be marked between elements of the following categories: Nouns, Noun Phrases and Pronouns. Elements of these categories are

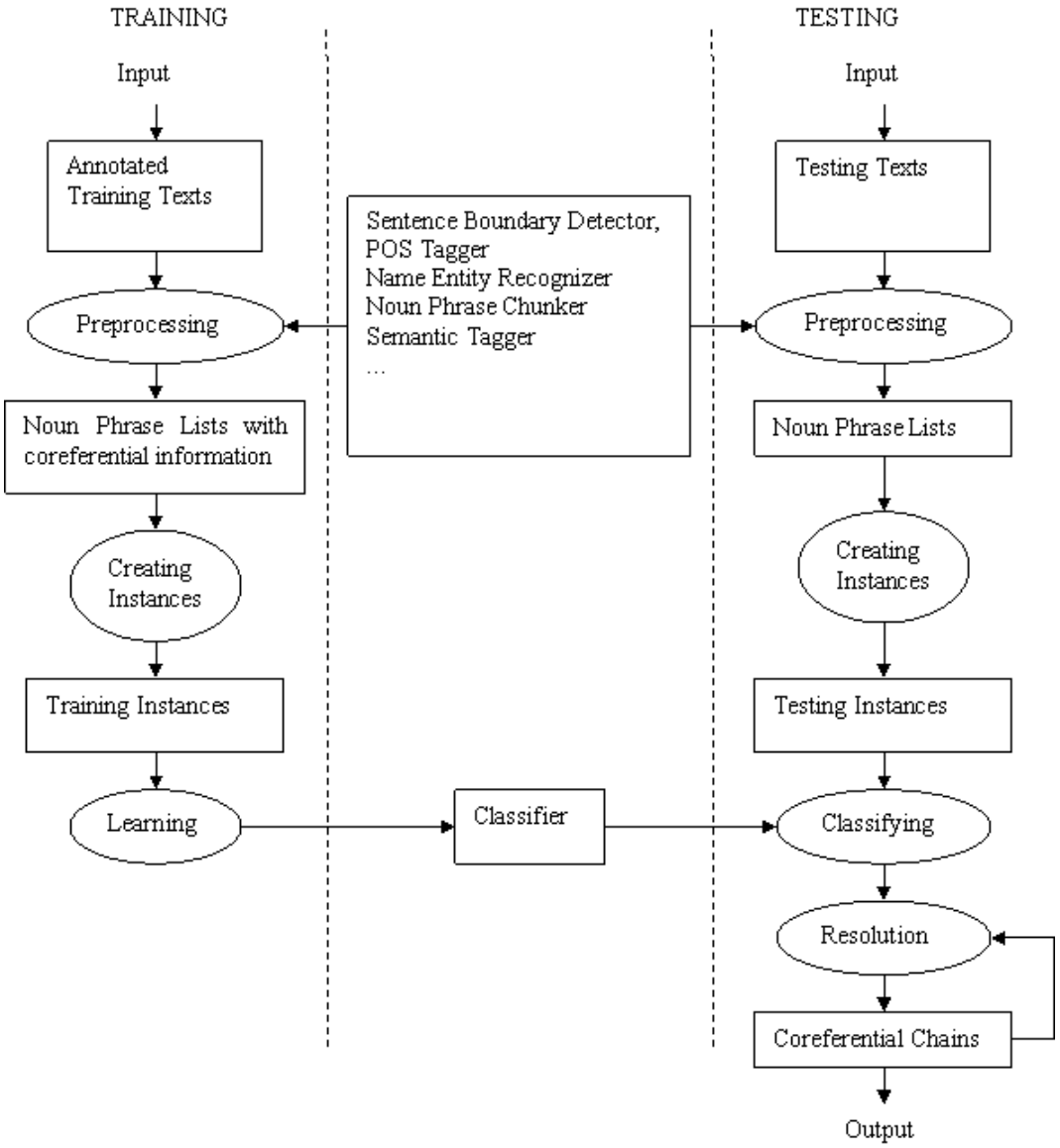


Figure 7-1: The framework of the coreference resolution system

called *markables*. Here pronouns include both personal and demonstrative pronouns, and with respect to personal pronouns, all cases, including the possessive. Dates (e.g. "January 21"), currency expressions (e.g. "$1.2 billion"), and percentages (e.g. "17%") are also considered noun phrases.

The annotation for coreference uses the SGML format. Markables in texts are marked up by an enclosing tagger <COREF> and </COREF>. A markable contains an attribute *ID*, which is the identification number of the current markable, and an attribute *REF* which is the ID of a coreferential NP. Generally, a markable is referred to its immediate antecedental NP and the whole coreferential link is established based on the symmetric and transitive properties of coreference. In addition to the two attributes, a markable could also have attributes like TYPE, MIN and STATUS. Table 7.1 gives a segment of an annotated text from the data set. The detailed definition of each attribution and the annotation scheme can be found in (Hirschman, 1998).

<s> <COREF ID = " 3 " TYPE = " IDENT " REF = " 4 " > QVC Network Inc. </COREF> , as expected , named <COREF ID = " 5 " TYPE = " IDENT " REF = " 1 " > Barry Diller </COREF> <COREF ID = " 7 " TYPE = " IDENT " REF = " 5 " MIN = " chairman " STATUS = " OPT " > <COREF ID = " 6 " TYPE = " IDENT " REF = " 3 " > its </COREF> chairman </COREF> and <COREF ID = " 8 " TYPE = " IDENT " REF = " 5 " MIN = " officer " STATUS = " OPT " > chief executive officer </COREF> . </s>

Table 7.1: A segment of an annotated text in the MUC data set

In MUC-6 and MUC-7, there are 30 "dryrun" documents for training, and 20 (MUC-7) and 30 documents (MUC-6) for "formal" testing. these documents were used in our experiments for the purpose of comparison with other coreference res-

olution systems. In addition, in MUC-6, there are a broad number of annotated texts that are annotated by the different participant sites. For better evaluation of the twin-candidate model in both antecedent selection and coreference resolution, 150 texts, with the document IDs starting with "891101" and "891102", were also utilized in our experiments.

7.1.2 Pre-processing Modules

The purpose of the preprocessing NLP modules is to determine the boundary of each NP in a text, and to provide necessary information for an NP for subsequent reference determination. As shown in Figure 7-1, these modules include Tokenization, Sentence-Boundary detection, Part-of-Speech tagging, Noun-Phrase chunking, Named-Entity recognition, and so on.

Early coreference resolution systems process input documents manually. Although human judgment can yield good results, it would be very time-consuming and may cost considerable human effort. Nowadays, with the development of sophisticated NLP tools, more and more practical coreference systems employ automatic approaches to pre-processing the documents. In our system, the pre-processing jobs are done all automatically using computational strategies.

Tokenization & Sentence Boundary Detection : In a raw document, punctuation marks or signs are often stuck with the preceding words. Some punctuation marks (e.g. "(","<","{", etc) are possibly confused with the mark-up tags in the annotation documents. The task of this module is to separate the individual words and marks in texts, and to identify the special marks and transfer them to corresponding escaping strings (e.g., in our system "(" becomes "LRB" and "{" becomes "LCB", and so on). Besides, in an input text, it is likely that a sentence is appended to the

preceding one without using any apparent separator. The module is also in charge of identifying the boundary of a sentence and inserting the separating tags (e.g. <s>) between sentences.

As an example, consider the following sentence,

(**Eg 7.1**) *. . . John Thrasher, Tower's top video buy, says he is "really pleased" with most of the test locations, but hastens to add:" I don't know how big of an investment that we as a retailer can invest in {another} video format." . . .*

After being processed, it becomes:

. . .<s> John Thrasher , Tower 's top video buy , says he is " really pleased " with most of the test locations , but hastens to add : " I do n't know how big of an investment that we as a retailer can invest in LCB another RCB video format . " <s> . . .

In our system the tokenization and sentence boundary detection are done just using a set of simple heuristic rules.

Part-of-Speech Tagging: Part of speech (POS) tagging is to tag each word in an input sentence with its most likely POS category. For example, consider the following sentence that has been tokenized:

(**Eg 7.2**) *Eastern Airlines executives notified union leaders that the carrier wishes to discuss selective wage reductions on Feb. 3 .*

The output of the POS tagging module is

(NNP Eastern) (NNP Airlines) (NNS executives) (VBD notified) (NN union) (NNS leaders) (IN that) (DT the) (NN carrier) (VBZ wishes) (TO to) (VB discuss) (JJ selective) (NN wage) (NNS reductions) (IN on) (NNP Feb.) (CD 3) (. .)

Here the names of the POS tags follow the POS guideline for the Penn Treebank Project (Santorini, 1990)

The tagging module in our system uses a statistics based approach on the HMM model by Zhou and Su (2000). The idea of using statistics for tagging and chunking goes back to (Church, 1988), who used corpus frequencies to determine the POS tagging sequence and the boundaries of noun phrases.

Given a token sequence $G_1^n = g_1 g_2 \dots g_n$, The goal of tagging is to find an optimal tag sequence $T_1^n = t_1 t_2 \dots t_n$ that maximizes

$$\lg P(T_1^n|G_1^n) = \lg P(T_1^n) + \lg \frac{P(T_1^n, G_1^n)}{P(T_1^n) * P(G_1^n)} \tag{7.1}$$

Then second item is the mutual information between T_1^n and G_1^n. In order to simplify the computation of this item, we assume that mutual information is independent, that is,

$$\lg \frac{P(T_1^n, G_1^n)}{P(T_1^n) * P(G_1^n)} = \sum_{i=1}^{n} \lg \frac{P(T_i, G_1^n)}{P(T_i) * P(G_1^n)} \tag{7.2}$$

Thus we have:

$$\begin{aligned} \lg P(T_1^n|G_1^n) &= \lg P(T_1^n) + \sum_{i=1}^{n} \lg \frac{P(T_i, G_1^n)}{P(T_i) * P(G_1^n)} \\ &= \lg P(T_1^n) - \sum_{i=1}^{n} \lg P(t_i) + \sum_{i=1}^{n} \lg P(t_i|G_1^n) \end{aligned} \tag{7.3}$$

In equation 7.3, the first item can be computed by applying chain rules, based on a backoff bi-gram model in which each tag is assumed to be probabilistically dependent on the previous tag. The second item is obtained by summing up the log probabilities of all the individual tags. The third item can be estimated by using the forward-backward algorithm recursively (Rabiner, 1989). The optimal tag sequence is found by maximizing the above equation over all the possible tag sequence, using the Viterbi algorithm.

Chunking: Chunking divides an input sentence into non-overlapping segments, and then identifies the category of the divided segments. Consider the sentence in Eg 7.2 again. The chunking result is:

(NP (NNP Eastern) (NNP Airlines) (NNS executives)) (VP (VBD notified)) (NP (NN union) (NNS leaders)) (SBAR (IN that)) (NP (DT the) (NN carrier)) (VP (VBZ wishes) (TO to) (VB discuss)) (NP (JJ selective) (NN wage) (NNS reductions)) (PP (IN on)) (NNP Feb.) (CD 3) (O (. .))

Text chunking can also be thought of as a tagging task by inserting brackets and labels into a POS sequence. Therefore, the same HMM model described for the POS tagging is also applicable to the chunking (Zhou and Su, 2000).

Given an input document, the module does chunking for all types of phrases. Among them, the noun phrase chunking is crucial in that coreference relations are built only on NPs in our study.

Named Entity Recognition: Named Entity (NE) Recognition (NER) is done based on the output of the POS and Chunking tagger. It is used to identify the entity names in a document and classify them into predefined semantic categories. As an

example, the output of the sentence of Eg 7.2 from the NER module is:

*(NP <**ENAMEX TYPE = " ORG " SUBTYPE = " Com:* "** > (**NNP Eastern) (NNP Airlines)** </**ENAMEX**> (NNS executives)) (VP (VBD notified)) (NN union) (NNS leaders)) (SBAR (IN that)) (NP (DT the) (NN carrier)) (VP (VBZ wishes) (TO to) (VB discuss)) (NP (JJ selective) (NN wage) (NNS reductions)) (PP (IN on)) <**TIMEX TYPE = " DATE "** > (**NNP Feb.) (CD 3)** </**TIMEX**> (O (. .))*

In fact, NER is quite similar to the task of text chunking, in the ways of identifying the boundary of the names in text and determining the corresponding category. Thus, the NER module (Zhou and Su, 2002) in our system also adopts the same HMM model as used in the POS and Chunking modules.

The NER module can recognize the NE categories defined in the MUC NE task (Chinchor, 1997): *person*, *organization*, *location*, *time* and *number*. More fine-grained distinction can be obtained for each category. For example, *organization* is divided into *company*, *government*, *institute*, etc, while *location* is divided into *region*, *country*, *city* and *water*, etc. For *time* and *number*, the subclasses include *day*, *month*, *year* as well as *money* or *percent*.

Named-entities determined by the NER module are merged into those found by the chunking module. If an NP overlaps with an NE, the boundaries of the NP are adjusted to subsume the NE.

The accuracy of the coreference resolution system depends, to a large extent, on the performance of the POS-tagger, the Chunker and the NER. In our system, the POS-tagging module obtains an accuracy of 97% and the Text-chunking module produces an F-measure of above 94%. Run on the MUC-6 and MUC-7 named-entity task, the NER module leads to an F-measure of 96.6% (MUC-6) and 94.1%(MUC-7),

which is significantly better than other systems applied to the same task.

7.1.3 Learning Algorithm

As described in the previous chapters, the twin-candidate model can use any discriminative learning algorithm that is capable of predicating a class label when given a feature vector. In our system, we employ the decision tree learning algorithm C5, an upgraded version of C4.5 (Quinlan, 1993), to train the classifiers.

The advantage of the decision tree learning algorithm is that a generated classifier can be easily interpreted by humans, and the importance of different features in question can be visualized. In fact, this algorithm is widely used in various coreference resolution systems (Aone and Bennett, 1995; McCarthy and Lehnert, 1995; Connolly et al., 1997; Soon et al., 2001; Ng and Cardie, 2002b; Strube and Mueller, 2003; Yang et al., 2004a). Here we would like to give a brief introduction to the algorithm C4.5.

In a decision tree, each node corresponds to a feature and each arc starting from the node represents a possible value of that feature. A leaf of the tree specifies the expected class label for the instance described by the path from the root to that leaf.

The basic idea behind C4.5 is to select the most "informative" features as the root and then iteratively expand its subtrees. The informativeness of a feature is calculated based on *entropy*, a measurement of the information conveyed by a probability distribution. Given a distribution $P = (p_1, p_2, \ldots, p_n)$, the entropy of P is defined as follows:

$$I(P) = -(p_1 * \log(p1) + p_2 * \log(p_2) + \ldots + p_n * \log(p_n)) \tag{7.4}$$

Suppose a set of instances, T, is partitioned into disjoint classes $C_1, C_2, \ldots, C_k$. Info(T), the information needed to identify the class of an element of T is the entropy of the probability distribution of the partition $(C_1, C_2, \ldots, C_k)$:

$$\text{Info}(T) = I(\frac{|C_1|}{|T|}, \frac{|C_2|}{|T|}, \ldots, \frac{|C_k|}{|T|}) \tag{7.5}$$

If we first partition T, on the basis of the value of a feature F, into sets T_1, T_2, ..., T_n, then the information needed to identify the class of an element of T becomes the weighted average of the information needed to identify the class of an element of T_i, i.e. the weighted average of Info(T_i):

$$\text{Info}(F,T) = \sum_i \frac{|T_i|}{|T|} * \text{Info}(T_i) \tag{7.6}$$

The information gain due to the feature F is calculated as follows:

$$\text{Gain}(F,T) = \text{Info}(T) - \text{Info}(F,T) \tag{7.7}$$

It represents the difference between the information needed to identify an element of T, and the information needed to identify an element of T after applying F. The information gains are used to rank the features. The decision trees are built in the way that each node is the feature with the largest gain, among those not yet considered in the path from the root.

In a learned decision tree, each leaf is associated with the correct times and the incorrect times of the corresponding classification in the training data. We can use this information to estimate the confidence value of a classification result, based on the following smoothed ratio:

$$CF = \frac{c+1}{t+2} \tag{7.8}$$

where c is the number of correct instances and t is the total number of instances stored in the corresponding leaf node.

7.2 Evaluation and Discussions

In our study we evaluated the twin-candidate model in two steps. First, we examined the efficacy of the twin-candidate model in identifying correct antecedents for given

anaphors. Second, we investigated the capability of the twin-candidate model in coreference resolution where the resolution is imposed on every encountered NP, no matter whether it was anaphoric and not. The detailed results and discussion for each step will be given the next two subsections, respectively.

7.2.1 Antecedent Selection

Experimental Setup

Our experiments first evaluated the twin-candidate model in antecedent identification for anaphors. In our study we considered the following types of anaphora:

- PRON: the third person pronouns[1] : "she", "he", "it", "they" and their morphologic variants (like "her", "his", "him", "its", "itself", "them"...). We further discriminated between two subtypes of pronouns:
  - P-PRON: the third person pronouns with male or female gender, i.e., "she", "he", which usually have the semantic category of "Human".
  - N-PRON: the third person pronouns with neutral gender, i.e., "it", "they", which usually have no specific semantic category.
- DET : the definite noun phrases that start with the definite article "the".

We trained and tested the four types of anaphora (PRON, P-PRON, N-PRON, DET) with separated classifiers. For training, we used the 150 documents from the MUC-6 annotation collection, while for test we used 50 standard MUC-6 (30) and MUC-7 (20) "formal-testing" documents. The anaphors were those markables that have a preceding NP in their respective annotated coreferential chains. As the current

[1]In texts, first and second person pronouns usually refer to the current speakers or hearers. Their antecedents usually can be easily identified compared with third person pronouns.

experiments only focused on antecedent selection for anaphors, we used the success rate as the evaluation metric which is defined as follows:

$$Success = \frac{\text{the number of anaphors being correctly resolved}}{\text{the total number of anaphors to be resolved}} \tag{7.9}$$

Here an anaphor is considered "correctly resolved" if the found antecedent is in the coreferential chain of the anaphor.

Given a pronominal anaphor, the distance between the immediate antecedent and the anaphor is usually short, predominantly (98% for our data set) limited to only one or two sentences as found in (McEnery et al., 1997). Therefore, for PRON resolution (also, N-Pron and P-Pron), we took as the candidates the markables that occurred in the current or in the preceding two sentences of the anaphor, restricted to those between the anaphor and its closest non-immediate coreferential NP. Besides, the markables with mismatched number, gender and person agreements were removed from the candidate set in advance. In total, we got 1020 PRON anaphors (635 N-Pron, 385 P-Pron) for training, and 442 ones (245 N-PRON, 197 P-PRON) for testing. The average number of candidates per anaphor in testing is about 10.

For DET resolution, the influence of distance is not as apparent as for the pronoun resolution. Therefore, for training, we took as the antecedent candidates all the preceding non-pronominal markables in the current and four sentences apart from the anaphor, while for testing, we used all the preceding markables, regardless of the distance, as the candidates. Totally, we had 835 DET anaphors in the training data, and 520 in the testing data. The average number of candidates per anaphor in testing is about 99.

Table 7.2 summarizes the statistics of the training instances and the class distribution. Note that for single-candidate model, the number of "1" instances is identical to the number of anaphors in the training data, since the model only uses the immediate antecedents to create the positive instances. Also, for the single-candidate

		N-PRON	P-PRON	PRON	DET
Single-Candidate	0 instances	619	665	1284	6991
	1 instances	635	385	1020	835
	class distribution	1 : 1.0	1 : 1.8	1 : 1.3	1 : 8.4
Twin-Candidate	01 instances	2240	1097	3337	9223
	10 instances	619	665	1284	6991
	class distribution	1 : 3.6	1 : 1.6	1 : 2.6	1 : 1.3

Table 7.2: The statistics for the antecedent selection task

model, the number of "0" instances is identical to the number of the "10" instances for the twin-candidate model, since these "0" and "10" instances both come from the non-coreferential candidates between the anaphors and their immediate antecedents.

As observed, for PRON resolution, the twin-candidate model does not show higher balance in class distribution than the single-candidate model (1:2.6 vs 1:1.3). It should be because for a pronominal anaphor, its immediate antecedent usually occurs in a short distance. Thus the number of the intervening non-coreferential candidates, as well as the resulting negative instances, is not too large. For DET resolution, nevertheless, the antecedents occur comparatively far from the anaphors. As a result, the negative instances considerably outnumber the positive ones (1:8.4). By contrast, the class distribution in the twin-candidate model can remain balanced (1: 1.3).

Resolution Results

Our experiment investigated the following six systems:

SC SC is a system based on the single-candidate model as described in Section 3.2.2. During training, given an anaphor, a training instance is formed by pairing the anaphor and each of its preceding candidates, until the immediate antecedent is reached. Each instance is associated with a set of features, which are the same as those defined for the twin-candidate model (Table 6.1), except that only one

set of features related to the single candidate (no inter-candidate feature) is included.

During resolution, for each anaphor, a test instance is formed for the anaphor and each candidate. The instance is sent to the learned classifier which will determine whether the candidate is the antecedent of the anaphor.

The score of a candidate is calculated based on the classification confidence CF:

$$Score(C_i) = \begin{cases} CF & : \quad C_i \text{ is classified as positive} \\ -CF & : \quad C_i \text{ is classified as negative} \end{cases} \tag{7.10}$$

Thus the higher the score, the more likely that a candidate is the antecedent. The candidate with the highest score is selected as the antecedent.

TC-EL TC-EL is a system based on the twin-candidate model. The system uses the linear "*Elimination*" resolution scheme (see Section 5.1.4) to select the antecedent from the candidate set. That is, in the system, the first two candidates are first compared. The less preferred one is eliminated, while the winner continues to be compared with third one. The process goes on until the last candidate is reached. The winner in the last comparison is selected as the antecedent.

TC-ELR Similar to TC-EL, the system is based on the twin-candidate model, using the linear elimination resolution scheme. The difference is that TC-ELR searches for the antecedent in the reverse order, from the last candidate toward the first one.

TC-ELM TC-ELM is also based on the twin-candidate model, but using the multi-round elimination resolution scheme (see Section 5.1.4). In each round, comparisons are held between consecutive candidates co-currently. The preferred candidates in one round will continue to be compared in the next rounds. The

process repeats until only one candidate remains, and this final winner is then selected as the antecedent.

TC-RB This system is based the twin-candidate model, using the "*Round-Robin*" resolution scheme as described in Section 5.1.4. In the system, comparisons are held between every two candidates. The wining/losing difference of a candidate is recorded and the one that wins the maximum number of competitors is selected as the antecedent.

TC-RBW Similar to TC-RB, this system also adopts the round-robin model, but uses the confidence value of the preference classification as the weight to increase/decrease the record of a candidate. The candidate with the highest record is selected as the antecedent (see Section 5.1.4).

All the classifiers in the systems were learned with default learning parameters, using the features listed in Table 6.1. As described in Chapter 6, to better capture the preference between candidates, a set of inter-candidate features is used which are calculated based on the features from the two competing candidates. To examine the utility of the inter-candidate features against their base features, we trained and tested each twin-candidate based system under the environment with or without the inter-candidate features present. Specifically, the following three feature sets were tried in the experiments:

All-Features Using all the features as listed in Table 6.1.

Base-Features A subset of *All-Features*. It includes all features but the four inter-candidate features (*inter_SameSent*, *inter_BetterSemSim*, *inter_BetterStrSim* and *inter_SemMagPron*).

InterCandi-Features A subset of *All-Features*. It contains similar features as *Base-Features*, except that the four inter-candidate features (*inter_SameSent*,

		SubType of PRON			PRON	DET
		N-PRON	P-PRON	OVERALL		
	SC	70.6	86.8	77.8	75.3	67.7
Base-Features	TC-EL	75.5	91.9	82.8	76.5	69.0
	TC-ELR	75.5	91.4	82.6	77.6	68.8
	TC-ELM	75.5	91.4	82.6	76.7	68.5
	TC-RB	75.5	91.4	82.6	77.6	68.7
	TC-RBW	75.9	91.4	82.8	77.1	68.8
All-Features	TC-EL	76.7	92.4	83.7	80.5	69.2
	TC-ELR	77.6	92.4	84.2	81.0	69.6
	TC-ELM	77.1	92.4	83.9	81.0	69.2
	TC-RB	77.1	92.4	83.9	80.3	69.2
	TC-RBW	77.1	92.4	83.9	81.2	69.4
InterCandi-Features	TC-EL	78.4	91.4	84.2	81.7	70.4
	TC-ELR	78.8	91.9	84.6	82.3	70.8
	TC-ELM	78.8	91.9	84.6	81.9	70.8
	TC-RB	78.8	91.9	84.6	81.7	70.4
	TC-RBW	78.8	91.9	84.6	81.9	71.0

Table 7.3: The success rates of different systems in antecedent identification for anaphora resolution

inter-BetterSemSim, *inter-BetterStrSim* and *inter_SemMagPron*) are used in place of their base features (*SameSent*, *SemSim*, *StrSim*, *SemSimPron*) of the two candidates.

The results of different systems are summarized in Table 7.3. Note that the column "OVERALL" shows the overall pronoun resolution results obtained by combining the results of N-PRON resolution and P-PRON resolution. This is different from the column "PRON", in which the results come directly from the classifiers trained and tested on the whole pronouns.

The first line of Table 7.3 is for the single-candidate based systems SC. For PRON resolution, SC obtains 75.3% *success*. If trained and tested for N-PRON and P-PRON separately, it achieves a higher overall *success* of 77.8% (70.6% N-PRON, 86.8%

P-PRON). These results are comparable to those reported by Kehler et al. (2004) (around 75%), who also used the single-candidate model to do pronoun resolution in the newswire domain (ACE data) using similar features. For DET resolution, SC yields a success rate of 67.7%.

The remaining blocks of Table 7.3 summarize the results of the five twin-candidate based systems, each on a different feature set. As we can see, all the five systems yield gains in the success rates as opposed to the single-candidate based system SC. In particular, systems under *InterCandi-Features* can produce the largest improvement: For PRON resolution, these systems significantly[2] outperform the baseline system in *success* up to 7.0% (8.2% for N-Pron and 5.6% for P-Pron). For DET resolution, they bring smaller but still significant improvement against the baseline by up to 3.3% in *success*. These results prove our claim that the twin-candidate model is more effective than the single-candidate model in antecedent selection for anaphors.

Figure 7-2 and Figure 7-3 illustrate the decision trees (top 4 levels) for PRON resolution, generated by the single-candidate model and the twin-candidate model (under *InterCandi-Features*) respectively. As the twin-candidate model uses a larger size of features, the tree output by the twin-candidate model is more complicated than the one by the single-candidate model.

From the two trees we can find that the twin-candidate model is able to avoid ties in the comparisons of the candidates. For example, the tree by the single-candidate model makes the reference decision only by checking whether a candidate is a pronoun, the first mention, or a named-entity in turn. If two candidates are both pronouns, they will have the same confidence value and the preference relationship cannot be determined. By contrast, for the tree by the twin-candidate model, if a candidate is a pronoun, the properties (e.g., pronoun, first-mention, or semantic similarity) of the competing candidate will be further examined to determine which one is more

[2]Throughout our experiments, significance was examined by using a paired t-test, with $p < 0.05$.

preferred.

When comparing the different resolution schemes (*elimination* vs *round-robin*), we find no significant difference between the success rates (less than 1.1% for PRON and 0.6% for DET) produced by these schemes. This goes against our belief that the *round-robin* scheme, which is more reliable than the *elimination* one, should lead to better results. One possible reason is that the classifier in our systems can identify the preference between the coreferential candidates and the non-coreferential ones with a high accuracy (above 92% as in our test). Therefore, using the simple linear search is capable of leading to the final antecedent as correctly as the round-robin search. 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 has a complexity of $O(N)$, instead of $O(N^2)$ as in the *round-robin*).

It is interesting to note that the success rates under column "OVERALL" are higher than those under column "PRON". That is, the separate resolution of N-Prons and P-Prons yields an overall result better than the direct resolution on the whole pronouns. One reasonable explanation is that the resolution of the two types of pronouns relies on different knowledge (discussed later). The learning algorithm, however, may be not powerful enough to automatically find out the different resolution regularities that are suitable for each type. As a matter of fact, recent research on decision-tree learning (Li and Liu, 2003) has suggested that the ensembles of cascading trees rooted by different features will give better classification than the directly learned tree. Our case could be thought of as manually assigning "ana_PronType" as the root feature, and then letting the learning algorithm construct the subtrees accordingly. Such a "divide-and-conquer" strategy is helpful for the learning algorithm to mine the resolution rules for different coreference phenomena.

```
candi_Pron = 1: 1 (350/32)
candi_Pron = 0:
:...candi_FirstNP = 1:
    :...candi_Name = 1: 1 (171/12)
    :   candi_Name = 0:
    :   :...ana_PronType = 1: 0 (4/2)
    :       ana_PronType = 2: 0 (76/29)
    :       ana_PronType = 3: 1 (75/15)
    :       ana_PronType = 4: 1 (52/6)
    candi_FirstNP = 0:
    :...candi_Name = 1:
        :...ana_PronType = 1: 1 (9/3)
        :   ana_PronType = 2: 1 (95/26)
        :   ana_PronType = 3: 0 (155/50)
        :   ana_PronType = 4: 0 (0)
        candi_Name = 0:
        :...ana_PronType = 1: 0 (56/5)
            ...
```

Figure 7-2: The decision tree generated for PRON resolution under the single-candidate model

```
inter_SameSent = 0:
:...candi_Pron_I = 0:
:   :...candi_Name_I = 1:
:   :   :...ana_PronType = 4: 1 (0)
:   :   :   ana_PronType = 3: 1 (313/22)
:   :   :   ana_PronType = 1: ...
:   :   :   ana_PronType = 2: ...
:   :   candi_Name_I = 0:
:   :   :...candi_FirstNP_I = 0: 1 (1905/37)
:   :       candi_FirstNP_I = 1: ...
:   candi_Pron_I = 1:
:   :...inter_SemMagPron > 190: 10 (17/1)
:       inter_SemMagPron <= 190:
:       :...ana_PronType = 1: 10 (17/1)
:           ...
inter_SameSent = 1:
:...candi_Pron_J = 1:
    :...candi_FirstNP_I = 0: 1 (250/13)
    :   candi_FirstNP_I = 1:
    :   :...candi_Name_I = 0: 1 (25/6)
    :       candi_Name_I = 1: 10 (8/2)
    candi_Pron_J = 0:
    :...candi_Pron_I = 1: 10 (309/19)
        candi_Pron_I = 0:
        :...inter_SemMagPron > 0: ...
            inter_SemMagPron <= 0: ...
```

Figure 7-3: The decision tree generated for PRON resolution under the twin-candidate model

Feature Analysis

The effects of different feature sets on antecedent identification are also demonstrated in Table 7.3. For PRON resolution and DET resolution, the systems with *InterCandi-Features* produce higher success rates than those with *Base-Features* (4.1∼5.2% for Pron, 1.4∼2.2% for DET) and *All-Features* (0.7∼1.4% for Pron, 1.4∼2.2% for DET). This supports our assumption that the inter-candidate features would be more indicative than their base features to represent the preference between competing candidates.

We compared the trees generated under different feature sets, and found that *Base-Features* results in more complicated trees than *InterCandi-Features*. Under *Base-Features*, the learning algorithm tends to compare all possible values of the two candidates' *StrSim*, *SemSim* or *SemSimPron* features. By contrast, under *InterCandi-Features*, the algorithm only needs to consider the value of the inter-candidate feature, i.e., *inter_StrSim*, *inter_SemSim* or *inter_SemMagPron*, which explicitly captures the comparison between their corresponding base features.

In our experiments we were also interested in the utility of the features for antecedent selection of each type of anaphora. For this purpose, we divided the features into groups, and then trained and tested on one group at a time. Table 7.4 and Table 7.5 show the feature groups and their results for pronoun (N-Pron and P-Pron) resolution and DET resolution, respectively. The features that lead to an empty decision tree are not listed in the tables.

From the tables, we see that features may play different roles in different types of anaphora resolution. For DET resolution, the string matching features and semantic features are the most indicative. For N-Pron resolution, the features *candi_FirstNP* and *SemSimPron* are the most important while for P-Pron resolution, the lexical features are the most effective by resulting in the success rate as high as using the

Feature Groups	N-Pron	P-Pron
candi_Pron + *candi_Def* + *candi_InDef* + *candi_Name*	57.1	91.3
candi_FirstNP	65.3	48.7
SemSimPron	61.6	48.2
inter_SemMagPron	61.2	52.3

Table 7.4: Results of different features for N-Pron and P-Pron resolution

whole feature set. The analysis of the current data revealed that most personal pronouns refer back to a personal pronoun or NE with the semantic category of human. That is, simply resolving a personal pronoun to some personal pronominal or NE candidate is sufficient to guarantee a high success rate for the current data set, and thus the other features were never used. For N-Pron resolution, however, the semantic category of the anaphor is not specified, and thus the resolution depends more on the syntactic or semantic knowledge.

From the tables we could also find that the inter-candidate features, when used alone, outperform their base features in most cases. For N-Pron and P-Pron resolution, the inter-candidate feature *inter_SemMagPron*, yields similar or higher *success* than the individual feature *SemSimPron*. Likewise, for DET resolution, the systems using inter-candidate features *inter_BetterStrSim* or *inter_BetterSemSim* also outperform the systems directly using *StrSim* or *SemSim*. All these findings further prove that the inter-candidate features can effectively represent the preference between candidates and help antecedent selection for the anaphora resolution task.

Learning Curves

In our experiments we were also concerned about how the training data size influences the anaphora resolution performance. For this purpose, we tested the resolution systems using different numbers of training documents. Figure 7-4 and Figure 7-5 plot the learning curves for PRON resolution and DET resolution, respectively. Each

Feature Groups	Success
candi_Pron + *candi_Def* + *candi_InDef* + *candi_Name*	12.8
FullStrMatch	41.7
HeadStrMatch	60.8
StrSim	59.8
inter_BetterStrSim	63.3
SemSim	52.3
inter-BetterSemSim	62.3

Table 7.5: Results of different features for DET resolution

success rate shown in the figures is the average of the results of three trials. Here we only compare SC and TC-RB. Similar results can be obtained for other twin-candidate based systems.

As shown in the figure, for PRON resolution, the single-candidate model obtains better results than the twin-candidate model when the size of the training data is small (below 15). This could be because the number of the features in the TC model is nearly double of that in the SC model. Therefore, TC would probably require more training data than SC to avoid the data sparseness problem. Fortunately, the TC model does not need too much training data to outperform the SC model. With above 30 documents, TC can lead to the success rates consistently higher than the SC model. The number of training documents to outperform SC is even less (below 10) for DET resolution. For the two types of anaphora resolution, both the SC model and the TC model reach the peak performance under around 80∼100 training documents. With more training documents, the performance tends to increase comparatively slow or even decrease slightly.

7.2.2 Coreference Resolution

In the last subsection we have demonstrated that the twin-candidate model is more effective than the single-candidate model in the antecedent selection for anaphora res-

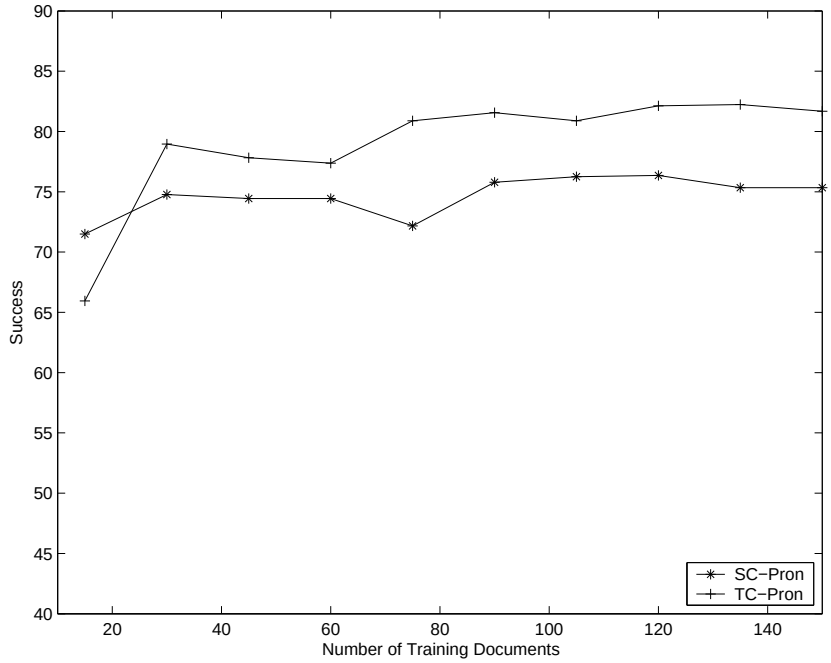


Figure 7-4: Learning curves of the single-candidate model and the twin-candidate model on PRON resolution

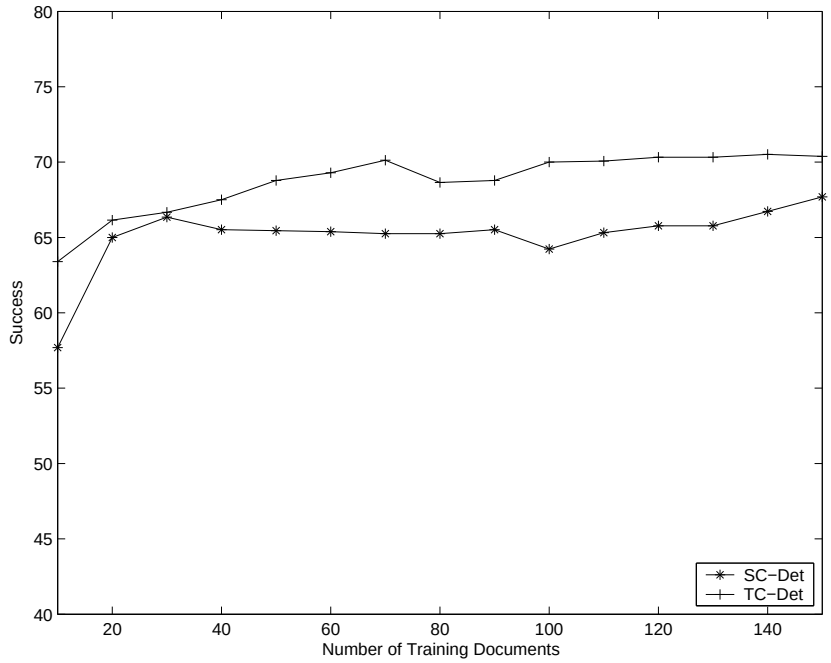


Figure 7-5: Learning curves of the single-candidate model and the twin-candidate model on DET resolution

olution. Now the concern is how the twin-candidate model performs in the coreference resolution task, where the anaphoricity of a given NP during resolution is unknown. In this part we will investigate the capability of the twin-candidate in coreference resolution.

Experimental setup

Consistent with the anaphora resolution task, in our experiments we also used the 150 annotated document from MUC-6 collection for training. As the coreference annotation guideline for MUC-6 is a little different from that for MUC-7, we only used the 30 MUC-6 "formal-testing" documents for test. In addition, for comparison with other work on coreference resolution, we also used the normal MUC-6 and MUC-7 data set for training and testing. In these experiments, the training was done on the 30 "dry-run" documents and the resolution was done on 30 and 20 "formal-testing" documents for MUC-6 and MUC-7 respectively. For evaluation, the recall and precision rates were calculated based on the evaluation metrics proposed by Vilain et al. (1995) (see Section 2.2.2).

As pronouns are anaphoric in general, we could simply use the basic twin-candidate model to select the antecedents for the pronouns encountered[3]. Specifically, in our study, the third-person pronouns were resolved using the N-PRON and P-PRON resolution systems described in previous section. The first-person or second-person pronouns were heuristically resolved to the closest pronoun of the same type or a speaker nearby, if any. For the non-pronouns, the resolution differs in the following systems to be tested:

SC SC is a coreference resolution system based on the single-candidate model (Section 3.2.2). The system uses the same feature set as the twin-candidate model,

[3]In our study, the pleonastic use of "it" was identified in advance using a set of predefined patterns, like "it + BE + ADJP", "MAKE it ADJP" and so on.

except that only one set of feature related to the single candidate is required (also, no inter-candidate features). For pronouns, the resolution is done in the same way as in PRON anaphora resolution, that is, the anaphor is resolved to the candidate with the highest score. For non-pronouns, the resolution is done using the "best-first" clustering strategy, that is, an encountered NP is resolved to the positive candidate that has the highest confidence value, or left unresolved if no positive candidate exists[4].

TC_AD TC_AD is based on the basic twin-candidate mode, in which non-anaphors are eliminated by an anaphoricity determination module in advance (Section 5.2.1). We built a supervised learning based AD module similar to the system proposed by Ng and Cardie (2002a)[5] . The AD classifier was trained on the same 150 annotation SGML documents.

TC_Filter TC_Filter is based on the basic twin-candidate model, but using the single-candidate classifier as the candidate filter (Section 5.2.2). The candidates are filtered by a classifier which is same as the one used in SC. Candidates that are considered as negative by the SC classifier are removed and the antecedent is selected from the remaining positive ones, if any, using the twin-candidate model.

TC_THRESH TC_THRESH is based on the basic twin-candidate model, using a threshold to discard the low-confidence comparison results between candidates (see Section 5.2.3). In the system, if no candidate has a positive score, the

[4]For pronoun resolution, we could also resolve an anaphor to the best *positive* candidate, instead of the best candidate regardless of the class. That, as tested in our experiments, led to a trade-off between recall and precision but not much difference in the overall F-measure.

[5]In the module, training instances are created for all NPs encountered in texts. An instance is labelled as positive if the NP is an annotated anaphor, or negative if not. Features are used to describe the properties of the NP and its relationships with candidate set. For resolution, a test instance is generated for an NP to be resolved. The instance is passed to the learned classifier which will then return a positive or negative label indicating the anaphoricity of the NP.

		MUC-6	MUC-7	150
Single-Candidate	0 instances	8920	10007	26908
	1 instances	1085	1012	3520
	class distribution	1 : 8.2	1 : 9.9	1 : 7.9
Twin-Candidate	01 instances	19632	17948	58085
	10 instances	8920	10007	26908
	00 instances	62097	78828	338736
	class distribution	1 : 2.2 : 7.0	1 : 1.8 : 7.9	1 : 2.2 : 12.6

Table 7.6: The statistics for the coreference resolution task

current NP is considered as non-anaphoric and left unresolved, otherwise the NP is linked to the antecedent selected as normal.

TC_NEW TC_NEW is based on the modified twin-candidate model (Section 5.2.4). In the system, the classifier determines whether the current NP is not an anaphor and no preference should be held between the two candidates under consideration. If so, both candidates receive a penalty of -1 or (-weight) as in their respective records. If no candidate has a positive score in the end, the current NP is considered as non-anaphoric and left unresolved. Otherwise it is linked to the antecedent selected as normal.

The definition of the access window of antecedent candidates for the non-pronouns were the same as for the DET anaphora resolution described in the previous subsection. The statistics of the training instances for each data set are summarized in Table 7.6.

Results

The results of the six systems on MUC-6 and MUC-7 are summarized in Table 7.7. All the classifiers in the systems were learned with default learning parameters, using all the features listed in Table 6.1. For the systems that run with a threshold, i.e.,

	MUC-6			MUC-7			150		
Experiments	R	P	F	R	P	F	R	P	F
Soon et al. (2001)	58.6	67.3	62.6	56.1	65.5	60.4	-	-	-
SC	71.2	63.8	67.3	68.8	62.8	65.6	68.8	63.6	66.1
TC_AD	63.6	67.7	65.6	60.3	65.1	62.6	60.9	71.6	65.8
TC_FILTER	71.3	63.9	67.4	69.2	63.1	66.0	69.2	63.5	66.2
TC_THRESH	71.5	60.5	65.5	69.0	62.1	65.4	72.7	61.6	66.7
TC_NEW	65.8	71.3	68.4	65.2	68.9	67.0	66.4	72.2	69.2

Table 7.7: The performance of different coreference resolution systems

TC_THRESH and TC_NEW, five-fold cross-evaluation was performed on the training data to select the optimal threshold value.

In the experiments we tested the twin-candidate based systems using different the resolution schemes (*elimination*, *round-robin*). However, like for the anaphora resolution, we found no significant difference between these schemes. In Table 7.7, we only listed the results using the round-robin (no-weight) resolution scheme.

The first line of Table 7.7 lists the results of the single-candidate based system by Soon et al. (2001). As introduced in Section 3.2.2, their systems were trained and tested on the same MUC-6 and MUC-7 data set using the similar learning framework and features. The system obtains 62.6% and 60.4% F-measure on the two data sets. In contrast, our baseline single-candidate system outperforms Soon et al. (2001)'s system in both recall and precision, and achieves an F-measure of 67.3% (MUC-6) and 65.6% (MUC-7).

The third line is for the system TC_AD. Compared with the baseline systems, TC_AD achieves a higher precision but a lower recall, resulting in an F-measure worse than that of SC. The analysis of the AD classifier reveals that it successfully identifies 79.3% anaphors (79.5% precision) for MUC-6, and 70.9% anaphors (76.3% precision) for MUC-7. That means, although the pre-processing AD module can partly avoid the wrong resolution of a non-anaphor, it eliminates many anaphors at the same

time, which leads to the lower recall for coreference resolution. In the experiments we attempted to adjust the learning parameters to obtain several classifiers with different capability in identifying positive anaphors, but this only resulted in tradeoffs between recall and precision, but with no effective resolution improvement in F-measure.

The fourth line is for the system TC_FILTER which uses the single-candidate classifier to filter candidates in advance. We can find that such a hybrid system improves both recall and precision against SC for MUC-6 and MUC-7. It lends us support that the twin-candidate gives more accurate ranking of the candidates than the single-candidate model for antecedent selection. However, as TC_FILTER is run based on the output of SC, the resolution performance is significantly subject to the results of latter. As a result, we can only observe a very slight improvement in F-measure (less than 0.4%) against SC.

The fifth line lists the results of the system TC_THRESH, which uses a threshold to block the low-confidence resolution. As shown, the system yields a higher recall, but unfortunately at the same time it leads to the lowest precision. As a result, the F-measure is even lower than the baseline systems. Such a pattern of higher recall and lower precision indicates that using a threshold can reduce, to some degree, the risk of eliminating true anaphors, but it is too lenient to effectively block the resolution of non-anaphors.

The last line of Table 7.7 is for TC_NEW, which uses the modified twin-candidate models. Compared with the baseline systems and all the other twin-candidate based systems, TC_NEW produces large gains in the precision rates, which rank the highest among all the systems. Although the recall also drops at the same time, the increase in the precision compensates for it well; we observe an F-measure of 68.4% for MUC-6 and 67.0% for MUC-7, significantly better than the single-candidate based systems and all the other twin-candidate based systems. These results suggest that with our modified framework, the twin-candidate model can effectively identify non-anaphors

Experiments	R	P	F	comments
SC_Neg	52.5	82.3	64.1	Using non-anaphors to create negative training instances
TC_BestAD	74.9	74.8	74.9	Using the "perfect" Anaphoricity Determination module

Table 7.8: The coreference resolution performance of other baseline systems

and block their invalid resolution, without affecting the accuracy of the antecedent determination for anaphors.

Table 7.7 also lists the results of the systems trained on 150 documents. The similar performance patterns can be observed for these systems as when trained on 30 training documents. Especially, on the 150 documents, TC_NEW yields an F-measure of 69.2%, which is higher than on 30 documents, and is also significantly better than all the other systems based on either the single-candidate model (3.1%) and the twin-candidate model (2.5% $\sim$ 3.4%). In the following analysis, we will focus on the results trained on this larger data set.

Other Baselines

Table 7.8 gives the results of some other baseline systems. As described, our modified twin-candidate model makes use of the candidates of non-anaphors to create "00" instances. Can the non-coreferential pairs formed by the non-anaphors and their preceding NPs, if incorporated, also help the single-candidate based system?

To answer this question, in the single-candidate model, we added the negative training instances formed by the non-anaphors into the training set, and then learned a new classifier to do coreference resolution. The first line of Table 7.8 shows the results of such a system, SC_Neg. Against SC, although the new system achieves gain in precision, it at the same time has a large loss in recall. As a result the overall F-

Experiments	R	P	F	comments
TC_NameAlias	27.4	83.4	41.3	only "NameAlias"
TC_Appositive	2.4	43.0	4.6	only "Appositive"
TC_StringMatch	38.2	68.8	49.1	only string matching Features ("HeadStrMatch", "FullStrMatch", "StrSim", "inter_BetterStrSim")
TC_SemSim	7.7	42.4	13.1	only "SemSim"
TC_BetterSemSim	8.2	41.6	13.8	only "inter_BetterSemSim"
TC_StrSim	37.4	69.0	48.5	only "StrSim"
TC_BetterStrSim	42.5	56.0	48.3	only "inter_BetterStrSim"
TC_BaseF	63.5	74.5	68.6	Using *Base-Features*
TC_InterCandiF	68.8	68.0	68.4	Using *InterCandi-Features*

Table 7.9: The coreference resolution performance with different features

measure was even lower (2.0%) than SC. The degradation of the performance may be because adding the negative instances intensifies the skewness of the class distribution in the training set (up to 1:80 as tested). Such unbalanced training instances would adversely affect the classifier learning[6].

For TC_AD, we used a learning-based AD module to determine the anaphoricity of an encountered NP. As the performance of TC_AD is subject to the AD module, one concern is what is the upper-bound performance of TC_AD, when running on a "perfect" AD module capable of determining anaphoricity with 100% accuracy. For this purpose, we run TC_AD only on the NPs that are marked as anaphors in the annotated texts. As listed in the last line of Table 7.8, such a system, TC_BestAD, can produce an F-measure of 74.9%, higher than all the twin-candidate based systems

[6]Our previous work (Yang et al., 2004c) suggested that by selecting proper pairs of non-anaphors and candidates in the training instances creation, the system performance would be possibly improved for the single-candidate model.

in Table 7.7, which suggests that the TC_AD has potential to get further improved if a more accurate AD module is available.

Features

Table 7.9 summarizes the results of TC_NEW when trained on different features. When used alone, only the features related to name-alias, appositive, and string and semantic similarity are effective, while the others produce an empty classifier.

In contrast to the anaphora resolution, the inter-candidate features seem not superior to their base features for coreference resolution. When used alone, the features *SemSim* and *StrSim* are able to obtain an F-measure similar with or higher than their corresponding inter-candidate features *inter_BetterSemSim* and *inter_BetterStrSim*. When used in combination, the system with only base features but no inter-candidate features (TC_BaseF) also slightly outperforms the system in which the inter-candidate features are included in place of their base features (TC_InterCandiF). This should be due to the fact that in coreference resolution, the values of the base features (*SemSim*, *StrSim*) are also informative: they act as constraint factors to block the resolution of a non-anaphor that has a low string or semantic similarity with the candidates. Therefore, simply using inter-candidate features without their base features is not sufficient to make correct coreference resolution. This can explain why the inter-candidate features result in lower precision than their base features, as shown in the table. In spite of this, inter-candidate features can still be helpful for coreference resolution when used together with the base features: we see that the system with the whole features (i.e., TC_NEW) can improve the performance of TC_BaseF by 0.6% F-measure.

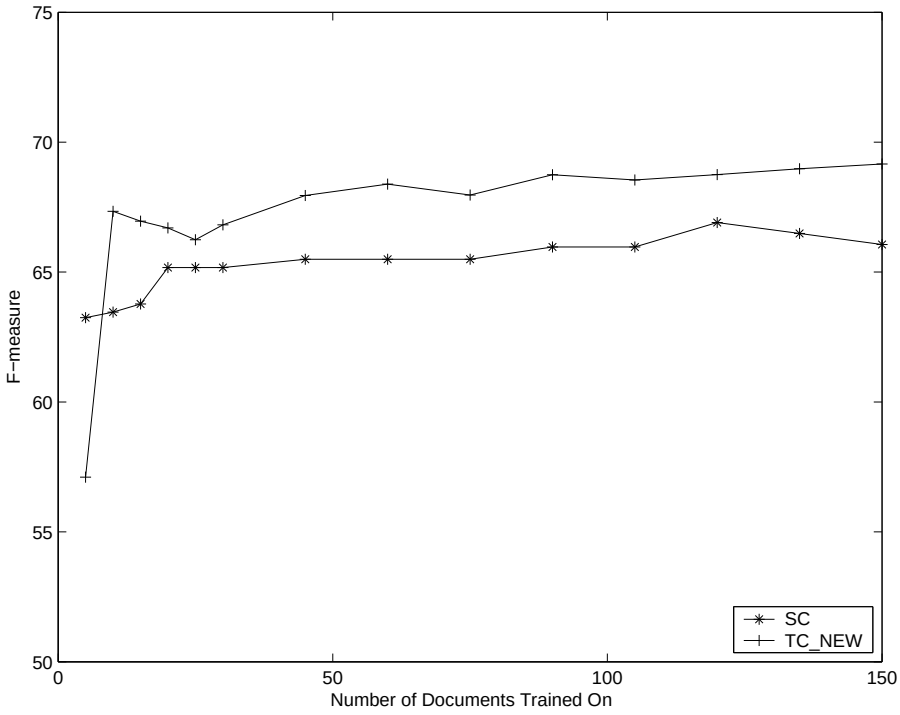


Figure 7-6: Learning curves of the coreference resolution systems

Learning Curves

In our experiments we were interested to evaluate the resolution performance of TC_NEW under different sizes of training data. Figure 7-6 plots the learning curve for the system TC_NEW as well as the single-candidate based system SC. The F-measure is averaged over three random trials trained on 5, 10, 15, ..., 135 and 150 documents. Consistent with the learning curves for the anaphora resolution task depicted in Figure 7-4 and 7-5, TC_NEW does not perform better than SC with small training data (less than 5 documents), but it can consistently outperform the latter when more data is available. The system achieves its peak performance with around 50 documents, and maintains this level with an increase of the training data.

Comparison between the TC-based systems

To provide a deeper comparison between the TC-based system, in Figure 7-7 we plotted the variant recall and precision rates that the four twin-candidate based systems were capable of producing when trained on the 150 documents. For TC_THRESH and TC_NEW, we obtained different recall and precision rates by adjusting the thresholds, while for TC_AD and TC_FILTER, we obtained them by adjusting the output of the AD and the filter classifier[7]. In line with the results in Table 7.7, the system TC_AD tends to obtain higher precision but lower recall, while the system TC_THRESH tends to obtain higher recall but lower precision. Comparatively, the system TC_NEW produces even recall and precision. For the range of recall (precision) in which the four systems overlap, TC_NEW always yields higher precision (recall) than the other systems. This figure demonstrates that the systems with our modified twin-candidate model is more reliable for coreference resolution than those with the other solutions.

As mentioned, systems TC_THRESH and TC_NEW have an adjustable threshold parameter. It is interesting to evaluate the influence of threshold values on the resolution performance. In Figure 7-8 we compare the different recall and precision rates of the two systems, with thresholds ranging from 65 to 100.

For TC_THRESH, when the threshold is low, the recall is almost 100% while the precision is quite low. In such a case, all the markables, regardless of anaphoric or non-anaphoric, will be resolved. As a consequence, all the occurring markables in a document tend to be linked together. In fact, the effective range where the threshold leads to an acceptable performance is quite limited. The threshold only works when it is considerably high (above 0.95). Before that, the precision remains very low (less than 40%) while the recall keeps going down with the increase of the threshold.

By contrast, for TC_NEW, both the recall and the precision rates vary little unless

[7]In our experiments we do this by setting different "misclassification-cost" parameters of C5.

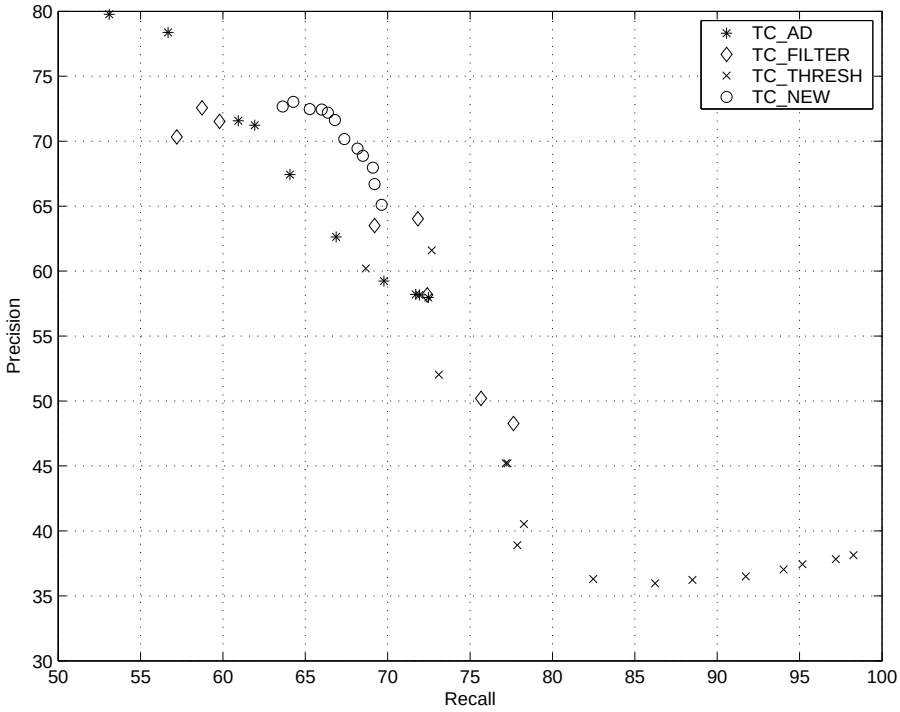


Figure 7-7: Various recall and precision rates for the twin-candidate based systems

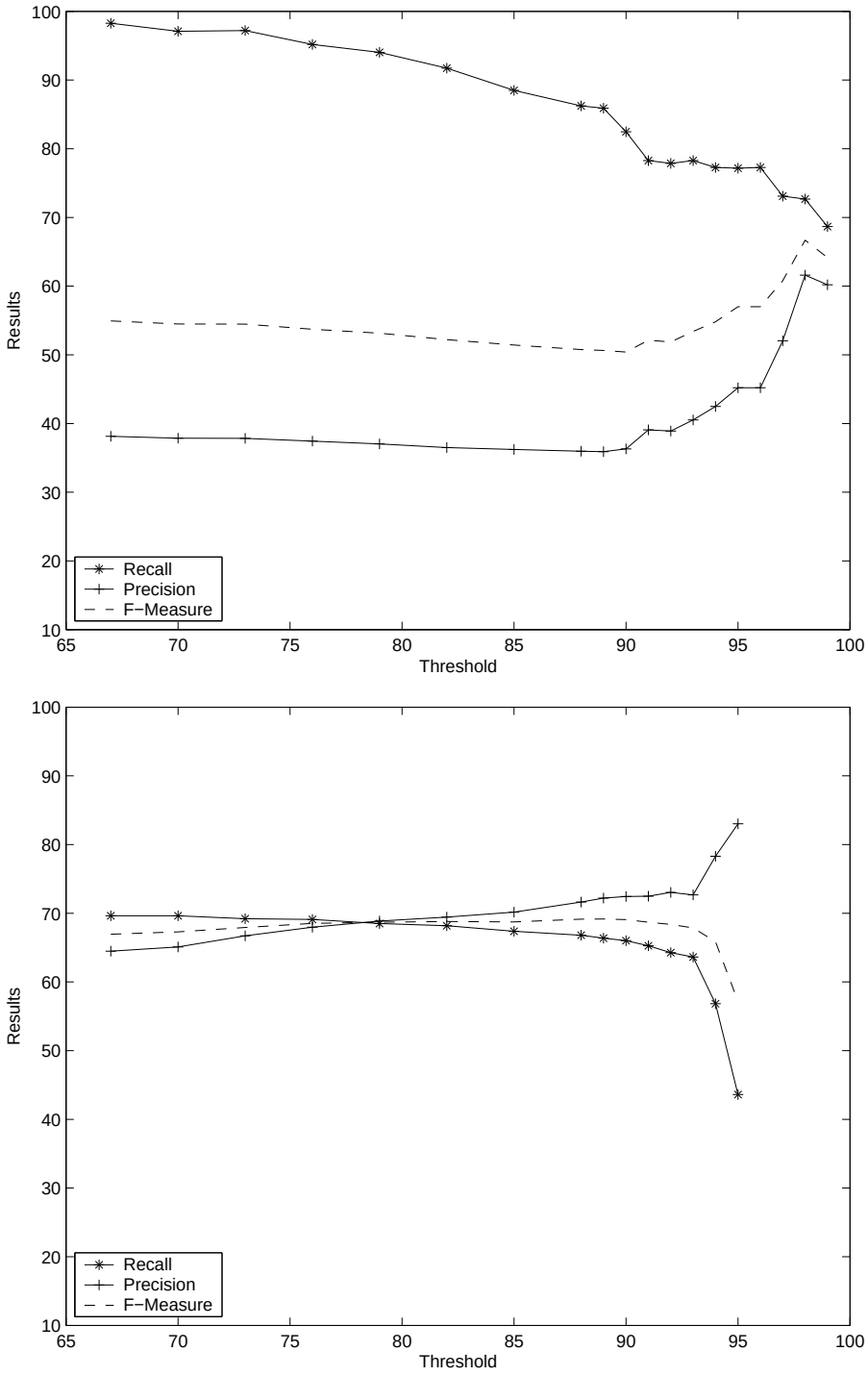


Figure 7-8: Influence of different threshold values on the coreference resolution performance

the threshold is extremely high. We could observe a very flat curve for F-measure before it starts to degrade. That means, the threshold does not impose much influence on the resolution performance of TC_NEW. This is because in the modified framework, the cases of non-anaphors are determined by the special class label "00", instead of the threshold as in TC_THRESH. The purpose of using a threshold in TC_NEW is not to identify the non-anaphors, but to improve the accuracy of class labelling. Indeed, from the figure, TC_NEW can obtain a good result without using any threshold. This comparison further confirms that our modified learning framework performs more reliably than the solution of using a threshold.

Comparison with Related Work

To our knowledge, our work is the first one to do coreference resolution using the twin-candidate model. The research efforts of Connolly et al. (1997) and Iida et al. (2003), as far as we know, are the only ones that attempt to employ such a model for anaphora resolution. As introduced in Chapter 3, Connolly et al. (1997)'s work included a limited number of features such as lexical types, grammatical role, recency and number/gender/semantic agreement. Their system obtained a comparatively low *success* for pronoun resolution (55.3%) and definite NP resolution (37.4%), on a set of selected news articles. Iida et al. (2003)'s work focused on Japanese zero-anaphora resolution. Their system incorporated centering features to capture the contextual knowledge, such as the rank of the candidates in a salience reference list. The system achieved a *success* around 70% on a data set drawn from a corpus with newspaper articles. Both of their works were evaluated on uncommon data sets, which makes it difficult to compare their results with others.

For the single-candidate model, there exists much more work trained and tested on the common MUC-6 and MUC-7 data set. Fisher et al. (1995)'s system, RESOLVE, is one of the MUC-6 systems that are based on supervised learning. RESOLVE em-

ployed 27 domain-independent features, and 8 domain-specific features like "whether an NP refers to a unit or a subsidiary of a certain parent company?" The system reported an F-measure of 47.2% for the MUC-6 test data. Soon et al. (2001) described a learning based system that was the first one evaluated on both MUC-6 and MUC-7. As having been introduced, their system made use of a smaller set of 12 generic and domain-independent features, and achieved encouraging results of 62.6% and 60.4% for MUC-6 and MUC7, respectively. Later, Ng and Cardie (2002b) extended Soon et al. (2001)'s work in the aspects of clustering strategy, training instance selection and feature definition. Their system achieved an F-measure of 69.1% (MUC-6) and 63.4% (MUC-7). In their another work, Ng and Cardie (2002a) attempted to add the anaphoricity information of noun phrases to help coreference resolution, and reported an F-measure of 65.8% and 64.2% for MUC-6 and MUC-7, respectively.

7.3 Summary

In this chapter we gave a comprehensive evaluation on the twin-candidate model proposed in the thesis. At first we described the resolution framework based on which the coreference resolution systems are run. We introduced the corpora, the pre-processing modules and the learning algorithm that are used in the systems.

The evaluation in our experiments were done in two steps. First, we investigated the capability of the twin-candidate model in antecedent selection for the anaphora resolution task. We examined the resolution performance of two types of anaphora: third-person pronominal anaphora and definite-NP anaphora. We found that for both types of anaphora, the twin-candidate model leads to better *success* than the single-candidate model. We also compared the twin-candidate systems with different resolution schemes like elimination or round-robin, but found not much difference between them. In addition, we explored the impact of training size on the resolution

performance, and found that the twin-candidate based model consistently outperforms the single-candidate one with a moderate size of training data. These findings support our assumption that the twin-candidate model is more effective than the single-candidate model in antecedent identification for anaphors.

Having shown the success in anaphora resolution, we were further concerned about how the twin-candidate model works in coreference resolution where the anaphoricity of an encountered NP is unknown. The results indicate that our system achieves significantly better results than the system based on the single-candidate model. A more detailed analysis of the twin-candidate based systems further proves that the system with our modified twin-candidate model is more reliable for coreference resolution than the twin-candidate based systems using other solutions.

In the experiments we also examined the utility of the features in the twin-candidate models. We found that for anaphora resolution, the inter-candidate features are more indicative, either when used in isolation or in combination, than their base features in identifying the antecedents, while for coreference resolution, these features do not show apparent superiority.

In the next chapter, we will give a conclusion of the work in our study.

Chapter 8

Conclusions

The purpose of our thesis is to find an effective learning model for the coreference resolution problem. The traditional single-candidate model is based on the assumption that the reference between an anaphor and an antecedent candidate is independent of the other candidates. However, for coreference resolution, the selection of the antecedent is determined by the preference between the competing candidates. The single-candidate model, which does reference determination by considering only one individual candidate at a time, cannot accurately capture the preference relationship between competing candidates. In addition, the single-candidate model would probably result in several positive candidates for a given anaphor. How to link the anaphor to a proper candidate becomes a problem and is often done in ad-hoc manners.

The main contribution of this thesis is that it presents a twin-candidate model that can overcome the above limitations of the single-candidate model. The remainder of the chapter will summarize and highlight the significance of the work that has been discussed in the previous chapters, and will discuss some potential directions for extending this work.

8.1 Main Contributions

With an aim to address the problems of the conventional single-candidate model, this thesis proposes, for the first time to our knowledge, a twin-candidate model to do coreference resolution. The main idea behind the twin-candidate model is to recast the antecedent selection as a preference classification problem. That is, the classification is done between two competing candidates to determine their preference as the antecedent of given an anaphor, instead of being done on one individual candidate to determine its reference with the anaphor. The thesis has the following contributions:

The construction of the twin-candidate model for antecedent selection

Chapter 5 gives the construction of the basic twin-candidate model for antecedent selection, including instance representation, training procedure and resolution procedure. In the model, a training instance is formed by the anaphor and two competing antecedent candidates. A training instance is labelled as "01" or "10" depending on which candidate is preferred to the other as the antecedent. A classifier is learned on the training instances, which is supposed to determine the preference between any two candidates of a given anaphor. The chapter proposes two possible resolution schemes, namely *elimination* and *round-robin*. In the elimination scheme, consecutive candidates are compared; the less preferred one is eliminated immediately while the winner continues for the subsequent rounds. The antecedent is the winner in the last comparison. In the round-robin scheme, a candidate is compared with every other candidate and the antecedent is the one that wins against the maximum number of competitors.

The efficacy of the twin-candidate model for antecedent selection is evaluated in Chapter 7. The experiments were done on the newswire domain, using MUC-6 and

MUC-7 coreference data set. The examination on different types of anaphora, Third-person Pronouns (PRON) and Definite-NP (DET), shows that the model achieves significantly better performance than the traditional single-candidate model, with the success rate increasing by up to 7.0% for PRON (8.2% N-Pron, 5.1% P-Pron) and 3.3% for DET. The learning curves indicate that the twin-candidate model can consistently outperform the single-candidate one with a moderate training size (30 documents for PRON and even less for DET). These results support our assumption that the twin-candidate model is more effective than the single-candidate model in identifying the correct antecedents for anaphors. The experimental results also show that there is not much difference in performance between the elimination scheme and the round-robin model, which suggests that the former is applicable to a practical system to make coreference resolution more efficient.

The application of the twin-candidate model in coreference resolution

Having shown the efficacy of the twin-candidate model in antecedent selection, now the issue is how to deploy the twin-candidate model to the coreference resolution task. The basic twin-candidate model aims to find the antecedent for an anaphor. However, in coreference resolution, it is often that an encountered noun phrase is non-anaphoric. Imposing the twin-candidate model to these NPs would lead to many false antecedents. To deal with this problem, Chapter 5 presents several possible solutions, for example, using an anaphoricity determination module to remove the non-anaphors in advance, or using a single-candidate base classifier to filter the candidates in advance, or using a threshold to block the resolution of an encountered NP if the classification confidence is not high enough. Nevertheless, all these solutions have their limitations. Our thesis proposes a modified twin-candidate model that makes use of non-anaphors to create a special set of training instances. The newly

learned classifier is capable of identifying the anaphoricity of the current NP and block the resolution by itself. Thus the model can do anaphoricity determination and antecedent selection at the same time.

Chapter 7 gives the evaluation on the different solutions to coreference resolution. The results indicate that the system using our modified twin-candidate model performs significantly better than the systems based on the traditional single-candidate model (up to 3.1% in F-measure) and the systems based on the basic twin-candidate model with the other solutions (2.5% $\sim$ 3.4%). The comparison between the learning curves shows that our system consistently outperforms the single-candidate based system when training on more than 5 documents. Furthermore, the in-depth analysis (e.g., under variant recall-precision combinations, or using different parameters) also reveals that our modified twin-candidate model is superior to the other solutions. These results indicate that our modified twin-candidate model can be reliably deployed for coreference resolution.

Knowledge representation in the twin-candidate model for coreference resolution

Chapter 6 explores the knowledge representation issue in the twin-candidate model. Our thesis proposes to utilize two types of knowledge for the coreference resolution task. The first type of knowledge is related to the individual candidate, describing their properties and their relationships with the anaphor, for example, "is the candidate a pronoun or a named-entity?", "How much do the candidate and anaphor match in strings or semantics?" By contrast, the second type of knowledge represents the relationships between the two competing candidates, for example, "between two candidates under consideration, which one has a higher string or semantic similarity with the anaphor?" Such inter-candidate knowledge can directly represent the

preference between the competing candidates, and thus can facilitate both preference learning and preference determination. In our study, all the adopted knowledge is domain-independent. The chapter gives a detailed description of these two types of knowledge in terms of features.

Chapter 7 also evaluates the utility of the features in the twin-candidate model for antecedent selection and for coreference resolution. We found that for anaphora resolution, by using the inter-candidate features in place of their base features brings gains in the success rate (up to 3.3% for N-Pron resolution and 2.5% as for DET resolution). This confirms our assumption that the inter-candidate features are more indicative than their base features for preference determination. However, for the task of coreference resolution, inter-candidate features do not show superiority over their base features. The reason is that the base features are also informative in blocking the resolution of non-anaphors, and thus simply using the inter-candidate features without the base features is not enough for coreference resolution. In spite of this, we observe that the inter-candidate features, when used together with their base features, can still improve the system performance. All these findings suggest that the inter-candidate features can be reliably used for both anaphora resolution and coreference resolution tasks.

8.2 Future Work

In addition to the contributions made by this work, a number of further contributions can be made by extending this work in new directions. Some of these potential extensions are discussed below.

8.2.1 Unsupervised or Weakly-Supervised Learning

In the current work we focus on a supervised learning method to coreference resolution. The baseline single-candidate model and the proposed twin-candidate model are both based on supervised learning.

In fact, as described in the literature review, so far there has been a proliferation of work attempting to solve coreference resolution problem by unsupervised (e.g. (Cardie and Wagstaff, 1999; Bean and Riloff, 2004)) or weakly-supervised methods (e.g. (Mueller et al., 2002; Ng and Cardie, 2003a)). Compared to the supervised learning approaches, these approaches require less, or even no, annotated data for rules learning, which can significantly reduce the human effort and are more adaptive on different domains. However, most of the current un(weakly)-supervised learning approaches also adopt the single-candidate model, that is, the reference determination is done by considering individual candidate only. For example, in Cardie and Wagstaff (1999)'s clustering algorithm, the distance metric is defined to calculate the compatibility between the anaphor and one candidate. Therefore, these approaches also face the same representation problem as in the supervised learning approaches based on the single-candidate model. That is, they cannot capture the preference relationship between candidates.

In our future work, we intend to investigate the use of the twin-candidate model in unsupervised learning approaches, for example, how to design the twin-candidate model that is capable of capturing the preference between candidates for unsupervised learning? How to make use of this model to do coreference resolution? How to represent the knowledge in the unsupervised learning based twin-candidate model? And how does such a twin-candidate model work under different impacting factors, compared with the single-candidate model, or compared with the twin-candidate model based on supervised learning?

8.2.2 Other Coreference Factors

One assumption behind the current twin-candidate model is that the preference relationship between two candidates is totally independent of other candidates. Thus the knowledge used in the twin-candidate model is restricted to the two competing candidates of a given anaphor. However, is there any other candidate existing that may affect the preference determination between two candidates?

In our previous work on coreference resolution (Yang et al., 2004a; Yang et al., 2004b; Yang et al., 2005a), we have found that the information of the antecedents of a candidate can help the decision whether the candidate is coreferential to the anaphor. Consider the following text, for example:

<s> [$_1$ Gitano] has pulled off [$_2$ a clever illusion] with [$_4$ [$_3$ its] advertising]. <s> <s> [$_5$ The campaign] gives [$_6$ its] clothes a youthful and trendy image to lure consumers into the store. <s>

Table 8.1: An example to demonstrate the necessity of antecedental information for pronoun resolution

In the above text, the pronoun [$_6$ its] has several antecedent candidates, i.e., [$_1$ Gitano], [$_2$ a clever illusion], [$_3$ its], [$_4$ its advertising] and [$_5$ The campaign]. Without looking back, [$_5$ The campaign] would be probably selected. However, given the knowledge that the company *Gitano* is the focus of the local context and [$_3$ its] refers to [$_1$ Gitano], it would be clear that the pronoun [$_6$ its] should be resolved to [$_3$ its] and thus [$_1$ Gitano], rather than other competitors.

To determine whether a candidate is the "focus" entity, we should check how the status (e.g. grammatical functions) of the entity alternates in the local context. Therefore, it is necessary to track the NPs in the coreferential chain of the candidate. For example, the syntactic roles (i.e., subject) of the antecedents of [$_3$ its] would indicate that [$_3$ its] refers to the most salient entity in the discourse segment.

The same problem also exists for non-pronoun resolution. As an individual candidate usually lacks adequate descriptive information of its referred entity, it is often difficult to judge whether the candidate and the anaphor are talking about the same entity simply from the pair alone. See the text segment in Table 8.2:

[$_1$ A mutant of [$_2$ KBF1/p50]], unable to bind to DNA but able to form homo- or [$_3$ heterodimers] , has been constructed. [$_4$ This protein] reduces or abolishes the DNA binding activity of wild-type proteins of [$_5$ the same family ([$_6$ KBF1/p50] , c- and v-rel)]. [$_7$ This mutant] also functions in vivo as a transacting dominant negative regulator:...

Table 8.2: An example to demonstrate the necessity of antecedental information for non-pronoun resolution

The co-reference relationship between the anaphor [$_7$ This mutant] and the candidate [$_4$ This protein] would be clear if the antecedent of the candidate is taken into consideration, i.e., [$_1$ A mutant of KBF1/p50].

Our previous work has suggested that incorporating the antecedental information of a candidate can effectively help the coreference determination between the candidate and the anaphor. However, this finding is based on the single-candidate model. Would such information be also helpful for the twin-candidate model? That is, for two candidates, C_i and C_j, should the candidates that are the antecedents of C_i and C_j be considered to determine the preference relationship between them? If so, how such knowledge is to be represented in the twin-candidate model? In our future work we would like to have a deep exploration on this issue.

Bibliography

ACE, 2000. *Entity Detection and Tracking - Phrase 1 ACE Pilot Study Task Definition*. http://www.itl.nist.gov/iad/894.01/tests/ace/.

C. Aone and S. W. Bennett. 1995. Evaluating automated and manual acquistion of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 122–129.

S. Azzam, K. Humphreys, and R. Gaizauskas. 1999. Using coreference chains for text summarization. In *Proceedings of the ACL Workshop on Coreference and its Applications*, pages 77–84.

A. Bagga and B. Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the LREC Workshop on Linguistic Coreference*, Granada, Spain.

B. Baldwin and T. Morton. 1998. Coreference-based summarization. In T. Firmin Hand and B. Sundheim, editors, *Proceedings of the TIPSTER Text Phase III Workshop*.

B. Baldwin. 1997. Cogniac: high precision coreference with limited knowledge and linguistic resources. In *Proceedings of the ACL97/EACL97 workshop on operational factors in practical, robust anaphora resolution*, pages 38–45, Madrid, Spain.

D. Bean and E. Riloff. 2004. Unsupervised learning of contextual role knowledge for coreference resolution. In *Proceedings of 2004 North American chapter of the Association for Computational Linguistics annual meeting*, pages 297–304.

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Learning Theory*, pages 92–100.

D. Bobrow. 1964. A question-answering system for high school algebra word problems. In *Proceedings of American Federation of Information Processing Societies (AFIPS) conference*.

B. Boguraev and C. Kennedy. 1997. Salience-based content characterisation of documents. In *proceedings of the ACL97/EACL97 workshop on intelligent scalable text summarisation*, pages 3–9, Madrid, Spain.

E. Breck, J. Burger, L. Ferro, D. House, M. Light, and I. Mani. 1999. A sys called qanda. In *Proceedings of the Eighth Text Retrieval Conference*, Gaithersburg, USA.

S. Brennan, M. Friedman, and C. Pollard. 1987. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 155–162.

J. Carbonell and R. Brown. 1988. Anaphora resolution: A multi-strategy approach. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 96–101.

C. Cardie and K. Wagstaff. 1999. Noun phrase coreference as clustering. In *Proceedings of the Joint Conference on Empirical Methods in NLP and Very Large Corpora*, pages 82–89.

D. Carter. 1987. *Interpreting Anaphors in Natural Language Texts*. Ellis Horwood, Chichester, UK.

E. Charniak and M. Johnson. 2005. Coarse-to-fine N-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180.

E. Charniak. 1972. Towards a model of children's story comprehension. Technical Report AI-TR 266, Artificial Intelligence Laboratory, MIT.

H. Chen. 1992. The transfer of anaphors in translation. *Literary and Linguistic Computing*, 7(4):231–238.

N. Chinchor. 1997. MUC-7 Named Entity task definition. In *Proceedings of the Seventh Message Understanding Conference*.

K. Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the 2nd Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas, USA.

M. Collins and N. Duffy. 2002. New ranking algorithms for parsing and tagging: kernels over discrete structures and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 263–270.

M. Collins and B. Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, pages 111–118, Barcelona, Spain.

D. Connolly, J. Burger, and D. Day, 1997. *A machine learning approach to anaphoric reference*, pages 133–144. New Methods in Language Processing.

I. Dagan and A. Itai. 1990. Automatic processing of large corpora for the resolution of anaphora references. In *Proceedings of the 13th International Conference on Computational Linguistics*, pages 330–332.

D. Fisher, S. Soderland, J. McCarthy, F. Feng, and W. Lehnert. 1995. Description of the UMass system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*.

R. Gaizauskas and K. Humphreys. 1997. Conceptions vs. lexicons: an architecture for multilingual information extraction. In *Proceedings of the Summer School on Inforamtion Extraction*, pages 28–43. Springer-Verlag.

N. Ge, J. Hale, and E. Charniak. 1998. A statistical approach to anaphora resolution. In *Proceedings of the 6th Workshop on Very Large Corpora*, pages 161–171.

B. Grosz, A. Joshi, and S. Weinstein. 1983. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual meeting of the Association for Computational Linguistics*, pages 44–50.

B. Grosz, A. Joshi, and S. Weinstein. 1995. Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

B. Grosz. 1977. The representation and use of focus in a system for understanding dialogs. In *Proceedings of the fifth International Joint Conference on Artificial Intelligence*, pages 67–76.

M. Halliday and R. Hasan. 1976. *Cohesion in English*. Longman English Language Series 9. Longman, London.

L. Hirschman. 1998. MUC-7 coreference task definition. In *Proceedings of the Seventh Message Understanding Conference*.

G. Hirst. 1981. *Anaphora in natural language understanding*. Springer Verlag, Berlin.

J. Hobbs. 1976. Pronoun resolution. Technical Report 76-1, Department of Computer Science, City University of New York, New York.

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

R. Iida, K. Inui, H. Takamura, and Y. 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"*.

D. Jurafsky and J. Martin. 2000. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition.* Prentice Hall.

M. Kameyama. 1997. Recognizing referential links: an information extraction perspective. In *Proceedings of the ACL97/EACL97 workshop on Operational factors in practical, robust anaphora resolution*, pages 46–53, Madrid, Spain.

R. Kantor. 1977. *The management and comprehension of discourse connection by pronouns in English.* Ph.D. thesis, Department of Linguistics, Ohio State University.

A. Kehler, D. Appelt, L. Taylor, and A. Simma. 2004. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proceedings of 2004 North American chapter of the Association for Computational Linguistics annual meeting*, pages 289–296.

A. Kehler. 1997a. Current theories of centering for pronoun interpretation: A critical evaluation. *Computational Linguistics*, 23(3):467–475.

A. Kehler. 1997b. Probabilistic coreference in information extraction. In *Proceedings of the second conference on Empirical Methods in Natural Language Processing*, pages 163–173.

F. Keller and M. Lapata. 2003. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3):459–484.

C. Kennedy and B. Boguraev. 1996. Anaphora for everyone: pronominal anaphora resolution without a parser. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 113–118, Copenhagen, Denmark.

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

J. Li and H. Liu. 2003. Ensembles of cascading trees. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM03).*

A. Lockman. 1978. *Contextual reference resolution.* Ph.D. thesis, Faculty of Pure Science, Columbia University.

X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. 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*, pages 135–142.

X. Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 25–32.

J. McCarthy and Q. Lehnert. 1995. Using decision trees for coreference resolution. In *Proceedings of the 14th International Conference on Artificial Intelligences*, pages 1050–1055.

J. McCarthy. 1996. *A Trainable Approach to Coreference Resolution for Information Extraction.* Ph.D. thesis, University of Massachusetts Amherst.

A. McEnery, 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.

G. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

R. Mitkov and P. Schmidt. 1998. On the complexity of anaphora resolution in machine translation. In Carlos Martin-Vide, editor, *Mathematical and computational analysis of natural language.* Amsterdam.

R. Mitkov, S. Choi, and R. Sharp. 1995. Anaphora resolution in machine translation. In *Proceedings of the Sixth International conference on Theoretical and Methodological issues in Machine Translation*, Leuven, Belgium.

R. Mitkov, K. Lee, H. Kim, and K. Choi. 1997. English-to-Korean machine translation and anaphor resolution. *Literary and Linguistic Computing*, 12(1):23–30.

R. Mitkov, R. Evans, C. Orasan, C. Barbu, L. Jones, and V. Sotirova. 2000. Coreference and anaphora: developing annotating tools, annotated resources and annotation strategies. In *Proceedings of the Discourse, Anaphora and Reference Resolution Conference (DAARC2000)*, pages 49–58, Lancaster, UK.

R. Mitkov. 1998. Robust pronoun resolution with limited knowledge. In *Proceedings of the 17th Int. Conference on Computational Linguistics*, pages 869–875.

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

N. Modjeska, K. Markert, and M. Nissim. 2003. Using the web in machine learning for other-anaphora resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 176–183.

T. Morton. 1999. Using coreference for question answering. In *Proceedings of ACL Workshop on Coreference and Its Applications*, pages 85–89, College Park, Maryland, USA.

MUC-6. 1995. *Proceedings of the Sixth Message Understanding Conference*. Morgan Kaufmann Publishers, San Francisco, CA.

MUC-7. 1998. *Proceedings of the Seventh Message Understanding Conference*. Morgan Kaufmann Publishers, San Francisco, CA.

C. Mueller, S. Rapp, and M. Strube. 2002. Applying co-training to reference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 352–359.

V. Ng and C. Cardie. 2002a. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING02)*.

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

V. Ng and C. Cardie. 2003a. Bootstrapping coreference classifiers with multiple machine learning algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

V. Ng and C. Cardie. 2003b. Weakly supervised natural language learning without redundant views. In *Proceedings of the North American chapter of the Association for Computational Linguistics annual meeting*, pages 94 – 101.

H. Ng, Y. Zhou, R. Dale, and M Gardiner. 2005. Machine learning approach to identification and resolution of one-anaphora. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 1105–1110, Edinburgh, Scotland.

V. Ng. 2004. Learning noun phrase anaphoricity to improve conference resolution: Issues in representation and optimization. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 151–158.

V. Ng. 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'05)*, pages 157–164.

M. Poesio and R. Vieira. 1998. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183–261.

M. Poesio, R. Mehta, A. Maroudas, and J. Hitzeman. 2004. Learning to resolve bridging references. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 143–150.

A. Popescu-Belis and I. Robba. 1998. Three new methods for evaluating reference resolution. In *Proceedings of the LREC Workshop on Linguistic Coreference*, Granada, Spain.

J. R. Quinlan. 1993. *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers, San Francisco, CA.

K. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE*, 77(2):257–285.

E. Riloff. 1996. An empirical study of automated dictionary construction for information extraction in three domains. *Artificial Intelligence*, 85:101–134.

C. Roberts. 2002. Demonstratives as definites. In K. Deemter and R. Ribble, editors, *Information Sharing*. CSLI, Stanford, CA.

H. Saggion and A. Carvalho. 1994. Anaphora resolution in a machine translation system. In *Proceedings of the International conference "Machine translation, 10 years on"*, Cranifield, UK.

B. Santorini, 1990. *Part-of-Speech Tagging Guidelines for the Penn Treebank project*, 3rd edition, June.

D. Shen, J. Zhang, G. Zhou, J. Su, and C. Tan. 2003. Effective adaptation of hidden markov model-based named-entity recognizer for biomedical domain. In *Proceedings of ACL03 Workshop on Natural Language Processing in Biomedicine*, Japan.

C. Sidner. 1978. The use of focus as a tool for the disambiguation of definite noun phrases. *Waltz*, pages 86–95.

C. Sidner. 1979. Toward a computational theory of definite anaphora comprehension in english. Technical report AI-TR-537, MIT, Cambridge, MA.

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

B. Srivinas and B. Baldwin. 1996. Exploiting supertag representation for fast coreference resolution. In *Proceedings of the NLP and IA conference*, pages 263–269, Moncton, Canada.

M. Stefik. 1995. *Introduction to Knowledge Systems.* Morgan Kaufmann, San Francisco, CA.

M. Strube and U. Hahn. 1999. Functional centering: Grounding referential coherence in information structure. *Computational Linguistics*, 25(3):309–344.

M. Strube and C. 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*, pages 168–175, Japan.

M. Strube, S. Rapp, and C. Mueller. 2002. The influence of minimum edit distance on reference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 312–319, Philadelphia.

M. Strube. 1998. Never look back: An alternative to centering. In *Proceedings of the 17th Int. Conference on Computational Linguistics and 36th Annual Meeting of ACL*, pages 1251–1257.

J. Tetreault. 1999. Analysis of syntax-based pronoun resolution methods. In *Proceedings of the 37th Annual Meeting of ACL*, pages 602–605.

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

R. Vieira and M. Poesio. 2000. An empirically based system for processing definite descriptions. *Computational Linguistics*, 27(4):539–592.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message understanding Conference (MUC-6)*, pages 45–52, San Francisco, CA. Morgan Kaufmann Publishers.

H. Wada. 1990. Discourse processing in MT: problems in pronominal translation. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING90)*, pages 73–75, Helsinki, Finland.

M. Walker. 1989. Evaluating discourse processing algorithms. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 251–261.

B. Webber. 1978. *A formal approach to discourse anaphora.* Ph.D. thesis, Department of Applied Mathematics, Harvard University.

Y. Wilks. 1973. *Preference Semantics.* Stanford AI Laboratory memo AIM-206. Stanford University.

Y. Wilks, 1975. *Preference semantics.* The formal semantics of natural language. Cambridge University Press.

S. Williams, M. Harvey, and K. Preston. 1996. Rule-based reference resolution for unrestricted text using part-of-speech tagging and noun phrase parsing. In *Proceedings of the Discourse Anaphora and Anaphora Resolution Colloquium (DAARC)*, pages 441–456, Lancaster, UK.

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

W. Woods, R. Kaplan, and B. Nash-Webber. 1972. The lunar science natural language information system: Final report. Technical report No 2378, Bolt Beranek and Newman Inc, Cambridge, Massachusetts.

X. Yang, J. Su, G. Zhou, and C. Tan. 2004a. Improving pronoun resolution by incorporating coreferential information of candidates. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 127–134, Barcelona.

X. Yang, J. Su, G. Zhou, and C. Tan. 2004b. An NP-cluster approach to coreference resolution. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 219–225, Geneva.

X. Yang, G. Zhou, J. Su, and C. Tan. 2004c. Improving noun phrase coreference resolution by matching strings. *Proceedings of the 1st International Joint Conference of Natural Language Processing (IJCNLP04), Lecture Notes in Computer Science*, 3248:22 – 31.

X. Yang, J. Su, and C. Tan. 2005a. Entity-based noun phrase coreference resolution. *Proceedings of the 6th Computational Linguistics and Intelligent Text Processing (CICLING05), Lecture Notes in Computer Science*, 3406:218 – 221.

X. Yang, J. Su, and C. Tan. 2005b. Improving pronoun resolution using statistics-based semantic compatibility information. In *Proceedings of the 43rd Annual*

Meeting of the Association for Computational Linguistics (ACL05), pages 165–172, Ann Arbor, USA.

X. Yang, J. Su, and C. Tan. 2005c. A twin-candidate model of coreference resolution with non-anaphor identification capability. In *Proceedings of the 2nd International Joint Conference of Natural Language Processing (IJCNLP05)*, Lecture Notes in Computer Science, pages 719 – 730, Jeju Island, Korea.

G. Zhou and J. 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.

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

G. Zhou and J. Su. 2004. A high-performance coreference resolution system using a constraint-based multi-agent strategy. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 522–528, Geneva.