A TRAINABLE APPROACH TO COREFERENCE RESOLUTION FOR INFORMATION EXTRACTION

A Dissertation Presented

by

JOSEPH F. MCCARTHY

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 1996

Department of Computer Science

To Amy

ACKNOWLEDGEMENTS

At the end of my graduate student career, I am nearly overwhelmed by three different feelings: exhaustion, relief and gratitude. I will merely acknowledge the first two, but want to elaborate a bit on the third.

My thesis advisor, Wendy Lehnert, has provided guidance and support throughout my work on this dissertation. Time and again she saw value in aspects of my research long before I did, and encouraged me to push on to the next stage.

I have also benefited from the input of my other committee members: Paul Utgoff, who offered many suggestions for improving this dissertation, but who also provided much-needed encouragement that some portions were fine as they were originally written; Paul Cohen, who asked tough questions but also offered useful suggestions for coming up with answers for them; and Gary Marcus, who recommended ways in which I could make this dissertation accessible to a broader audience.

I am grateful to my colleagues Stephen Soderland and David Fisher, who have always been generous with their time. Their careful readings of – and extensive feedback on – early (and late) versions of this dissertation helped me convert a rough draft into a nearly polished version in a very short time.

Former colleagues Ellen Riloff and Claire Cardie showed me that it was possible to work on three MUCs and still get a Ph.D. at the University of Massachusetts. As NLP Humor Director Emeritus, Ellen's support – both humorous and serious – helped me get through many difficult times.

Many other current and former colleagues offered their support throughout my time as a "gradual" student; I am particularly grateful to the members of my thesis support group – Malini Bhandaru, Carla Brodley, Jeff Clouse, Bob Crites, and Dorothy Mammen. Eric Brown, Jamie Callan, Amer Diwan, Brian Pinnette and David Yates have provided much support as well. I am also grateful to Priscilla Coe for her help on many occasions, especially during my frequent last-minute bustles.

I often fantasized about dropping out of the graduate program and seeking a "real" job. At one stage, I even discussed this prospect with a number of people. In addition to many of the people mentioned above, I am grateful to Charlie Dolan, Lisa Rau and Afshin Goodarzi for encouraging me to stick it out.

Thea Iberall provided much of the initial encouragement to start on my Ph.D., and has continued to be a valued source of support. Although I have often felt ambivalent about that initial encouragement, I feel unequivocal gratitude now that I have finished the degree.

John and Marilyn McCarthy, my parents, encouraged my inquisitiveness from an early age. While I am sure that at some stages of my life, this trait may not have been entirely well received, it certainly was an important factor in my academic path.

Jack and Mary Lou Gillstrom, my parents-in-law, have been tremendously supportive throughout my time in graduate school. I am very grateful for all their help during this period.

Meg and Evan McCarthy, my children, continually remind me what is really important in life, and have contributed to whatever semblance of balance I have been able to achieve over the past several years.

Amy McCarthy, my wife and closest friend, has really made all this possible. Whatever difficulties I have experienced during my work on this dissertation pale in

comparison to the sacrifices she has made to support me in this endeavor. For that, I am eternally grateful.

This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623.

ABSTRACT

A TRAINABLE APPROACH
TO COREFERENCE RESOLUTION
FOR INFORMATION EXTRACTION

SEPTEMBER 1996

JOSEPH F. MCCARTHY

This dissertation presents a new approach to solving the *coreference resolution* problem for a *natural language processing (NLP)* task known as *information extraction*. It describes a new system, named RESOLVE, that uses machine learning techniques to determine when two phrases in a text co-refer, i.e., refer to the same thing. RESOLVE can be used as a component within an information extraction system – a system that extracts information automatically from a corpus of texts that all focus on the same topic area – or it can be used as a stand-alone system to evaluate the relative contribution of different types of knowledge to the coreference resolution process.

RESOLVE represents an improvement over previous approaches to the coreference resolution problem, in that it uses a machine learning algorithm to handle some of the work that had previously been performed manually by a knowledge engineer. RESOLVE can achieve performance that is as good as a system that was manually constructed for the same task, when both systems are given access to the same knowledge and tested on the same data.

The machine learning algorithm used by RESOLVE can be given access to different types of knowledge, some portions of which are very specific to a particular topic area or *domain*, and other portions are more general or domain-independent. An ablation experiment shows that domain-specific knowledge is very important to coreference resolution – the performance degradation when the domain-specific features are disabled is significantly worse than when a similarly-sized set of domain-independent features is disabled.

However, even though domain-specific knowledge is important for coreference resolution, domain-independent features alone enable RESOLVE to achieve 80% of the performance it achieves when domain-specific features are available. One explanation for why domain-independent knowledge can be used so effectively is illustrated in another domain, where the machine learning algorithm discovers *domain-specific* knowledge by assembling the domain-independent features of knowledge into domain-specific patterns. This ability of RESOLVE to compensate for missing or insufficient domain-specific knowledge is a significant advantage for redeploying the system in new domains.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This dissertation presents a new approach to solving the *coreference resolution* problem for a *natural language processing (NLP)* task. When a new phrase, or *reference*, is encountered by a reader, the coreference resolution problem is to determine whether that phrase refers to something already known by the reader. The problem is often constrained by defining the set of things that are known by the reader to be the things already referenced by some preceding phrase in the same *discourse*. For the current work, a document or *text* will represent a single discourse, i.e., a sequence of sentences that focus on the same topic area.

The coreference resolution problem can be recast as a *binary classification problem*: given two phrases in a text, determine whether they refer to the same thing, i.e., whether they share the same *referent*. Those phrases that co-refer to the same thing are called *coreferent* phrases. For a pair of coreferent phrases, the phrase that occurs later in the discourse, is sometimes called an *anaphor*, and the phrase that occurs earlier in the discourse is sometimes called its *antecedent*.[1]

1.1 RESOLVE: A Trainable Coreference Resolution System

The new approach to solving the coreference resolution problem presented in this dissertation is implemented as RESOLVE, a coreference resolution system that learns how to classify pairs of phrases as coreferent or not coreferent. The system uses machine learning techniques to create a coreference classifier automatically from a set of training examples.

RESOLVE is designed to operate as a component of an *information extraction* system, an NLP system that automatically extracts narrowly specified information from a set of texts that are written about a specific topic area or *domain*. Information extraction systems differ from many other NLP systems in two important respects: they are designed to process real-world texts as opposed to texts specially constructed to test particular linguistic theories, and their goal is to extract certain kinds of information from a text rather than achieve a deep understanding of the text.

Within a single text, each new reference to an entity typically introduces new information about that entity, and several different entities are typically referenced, thus there exists a set of references and a set of potential referents. The coreference resolution component is used by an information extraction system to determine which pieces of information refer to the same entity, so that this information can be *merged* together as the system processes the text.

Most previous approaches to coreference resolution for large-scale NLP applications such as information extraction have employed manually encoded heuristics to make decisions about coreference relationships, or *links* among phrases. These manual approaches require knowledge engineers to do the following:

[1] The process of finding an antecedent for an anaphoric reference is sometimes called ***anaphor resolution*** [Sidner, 1979].

- Determine which aspects of each phrase must be identified in order to allow any coreferent relationships to be determined among the phrases – these aspects, or *features* are typically used in the antecedents of the rules.
- Determine how to combine these different aspects into individual rules – this process often involves determining which aspects combine to form positive evidence for coreferent relationships as well as determining which aspects must be included as exceptions, i.e., to form negative evidence for coreferent relationships.
- Determine the best ordering of the set of rules, and/or define some conflict resolution strategy that can be used when more than one rule applies to a given situation.

RESOLVE represents an important step forward in this knowledge engineering process. The use of a machine learning algorithm to combine and order the features eliminates the need for the knowledge engineer to determine such combinations and orderings manually. The knowledge engineer must still identify what kinds of information must be extracted from the individual phrases in order to determine coreferent relationships – and this is still a significant part of the knowledge engineering effort.

1.2 Goals of this Dissertation

Different theories have been proposed to account for the ways that phrases can refer to preceding phrases. Some of these theories have been implemented as computer programs. A few of these programs have even been tested on real-world texts. Most of these theories look at similar aspects, or features, of phrases, e.g., whether a phrase is a pronoun or definite reference, and the contexts within which phrases are found, e.g., which sentence a phrase occurs in or whether the phrase is the subject of that sentence. The differences among these theories lies primarily in their representation of these features and the relative importance they accord to each of the features.

The existence of an optimal set of features and an optimal arrangement or ordering of that set has not been conclusively established. Furthermore, a set and arrangement of features that works well in one domain may not work as well in other domains. A manually encoded algorithm for classifying coreferent phrases may need to be tuned manually for each new domain in which it is used. A *trainable* system may be able to determine the best features and best arrangement of features automatically, based on domain-specific training examples.

The hypothesis that motivates the work described in this dissertation can be stated as follows:

> *Machine learning techniques can be used to develop an effective coreference resolution system for information extraction.*

This hypothesis has been tested through a set of experiments in which RESOLVE was used as a stand-alone system; these experiments can be organized around two major themes which will be discussed in the two following sections. RESOLVE has also been used as a coreference resolution component within an information extraction system; the third section below describes an observation arising from this integration that provides additional, anecdotal evidence of the benefits of using machine learning for coreference resolution.

1.2.1 The Efficacy of a Machine Learning Approach

Machine learning techniques can be applied to virtually any problem that can be addressed by manual programming techniques. The real question is whether they can be applied *effectively* to a particular problem, i.e., whether a machine learning algorithm can learn to solve a problem at least as well as an existing algorithm, or a problem for which no other algorithm yet exists.

One of the contributions of this dissertation is to demonstrate that a machine learning approach to coreference resolution is as effective as other, manual approaches:

> *A coreference resolution system that uses machine learning techniques can achieve a level of performance comparable to a system that uses manually encoded heuristics.*

This issue is particularly important, since RESOLVE is intended to be used as part of an information extraction system. Applying machine learning techniques to the problem of coreference resolution is an interesting endeavor; however, since the system is designed as a component of a larger NLP system, it has to work as well as previous approaches.

This dissertation will show that when RESOLVE is given access to the same knowledge that was used in the manually encoded rules used by an existing coreference resolution system, it performs as well as the rule-based system. Since RESOLVE automates some of the knowledge engineering functions that were performed manually in developing the rule-based system, this approach represents an important advantage: less human knowledge engineering effort, but the same level of performance.

1.2.2 The Importance of Domain-Specific Knowledge

Information extraction systems require both knowledge of linguistic structure and knowledge of the world. Some of this knowledge is rather general, e.g., syntactic knowledge about subjects and direct objects or knowledge about the formats of proper names denoting people and companies, but some of the required knowledge is quite specific to a particular domain. An example of such *domain-specific* knowledge would be knowledge that can be used to extract information about how companies involved in a joint venture are related to one another, e.g., whether the companies are partners or whether one is the parent of the other.

Coreference resolution systems – and many other systems that perform specific natural language processing tasks – typically include both *domain-independent* knowledge and *domain-specific* knowledge. Each of the pieces of knowledge (or *features*) used by RESOLVE can be partitioned into one of these two categories. Having partitioned the features along this dimension, we show that

> *Domain-specific knowledge is important for coreference resolution in an information extraction system.*

In fact, the set of domain-specific features used for coreference resolution in one domain is more important to performance than *any* set of domain-independent features of the same size. Domain-independent features play an important role in coreference resolution, and in fact, there are some pairs of coreferent phrases that can be correctly identified using either domain-specific or domain-independent features. However, there are some pairs of coreferent phrases that simply cannot be correctly classified without domain-specific features.

1.2.3 RESOLVE as an Information Extraction System Component

RESOLVE was employed as a *stand-alone* system for the experiments mentioned in the previous two sections. The data upon which it was trained and tested did not come from an NLP system, but was generated by a special interface that permitted a person to annotate phrases in a text. Furthermore, the output of RESOLVE was not passed on to another NLP system component for further processing; instead, it was used to evaluate coreference performance.

RESOLVE has also been employed as an *embedded* coreference resolution component within an information extraction system that was developed under severe time constraints. Very little time was available for knowledge engineering, therefore RESOLVE had to rely upon its domain-independent features and one hastily constructed domain-specific feature.

An examination of RESOLVE's performance in this new domain illustrates a benefit to using machine learning for the coreference resolution problem: the system discovered a domain-specific pattern of coreference based on its domain-independent features. This learned rule helped to compensate for both the narrow definition of the single domain-specific feature and the noise present in the data that came from earlier stages of processing in the information extraction system.

The development of information extraction systems is often constrained by time, limiting the amount of knowledge that can be manually engineered for new domains. Noise resulting from other processing stages (prior to coreference resolution) is an unavoidable factor for a system that processes large numbers of real-world texts. The combination of these two aspects of real-world coreference resolution makes a machine learning approach to the problem especially appealing.

1.3 Organization of the Remainder of this Dissertation

Information extraction is an application area within the field of natural language processing. There are some aspects of information extraction that differentiate it from other areas of NLP research, and these novel aspects affect the coreference resolution task that RESOLVE was designed to perform. Chapter 2 will define this application area and describe some specific examples of information extraction tasks.

Information extraction imposes a number of constraints on the coreference resolution problem, simplifying some aspects of coreference classification. However, there are linguistic constructs not normally considered as referring expressions, e.g., indefinite references that encompass a broad class of entities, that must be linked by a coreference module in an information extraction system, which complicates the task. Chapter 3 will describe some of the issues that must be addressed by the coreference resolution component of an information extraction system; this chapter will also describe some related work on coreference resolution.

Applying machine learning techniques to a new problem is rarely a straightforward procedure. Decisions must be made about how the problem is represented, and a learning algorithm must be selected (or created). Chapter 4 will discuss these issues, describe how they are addressed in this work, and compare this work with other NLP applications of machine learning.

The data used for most of the experiments in this dissertation was collected by a special interface, CMI (the Coreference Marking Interface). Chapter 5 will describe this data in greater detail, and provide an overview of the way that the data was collected via this interface.

Predictive accuracy, the metric by which the performance of most machine learning algorithms are measured, does not adequately capture some important aspects of coreference resolution. Chapter 6 will discuss two other metrics, *recall* and *precision*,

which provide a better measurement of the performance of a coreference resolution system.

RESOLVE was given the same pieces of knowledge that were used by a set of manually encoded rules in the coreference module of an implemented information extraction system. Chapter 7 describes an experiment that compares the performance of these two systems. This experiment demonstrates that a machine learning approach can achieve the same level of performance as a manual approach. It also reveals some interesting behavior by a machine learning algorithm with respect to *unknown attributes*.

Chapter 8 describes the different features that constitute the domain-specific knowledge and the domain-independent knowledge used by RESOLVE in classifying coreferent phrases. When the eight domain-specific features were disabled, the performance of RESOLVE dropped by 20%, a significantly steeper degradation than was seen when any other randomly selected set of eight domain-independent features was disabled. This result highlights the importance of domain-specific knowledge to coreference resolution. The chapter concludes with a discussion about why RESOLVE is not able to find all coreference links among phrases even with access to all of its features, and a qualitative analysis of why the domain-specific features are so important.

Although domain-specific knowledge is important for coreference resolution, RESOLVE is still capable of finding many coreference links with only its domainindependent knowledge. One explanation for the effectiveness of domain-independent knowledge can be seen in the application of RESOLVE to a different domain than that used for its initial development – the machine learning algorithm used by RESOLVE combined domain-independent features to capture an important domain-specific pattern. Chapter 9 describes the issues involved in porting RESOLVE to a new domain, and provides a detailed analysis of the new, domain-specific pattern, and how this pattern was used in the new domain.

Chapter 10 will conclude the dissertation, highlighting its contributions to both machine learning and NLP, and describe some areas for future work.

CHAPTER 2

INFORMATION EXTRACTION

Understanding language is a process that comes quite naturally to most humans. Unfortunately, the naturalness with which we understand language makes it very difficult to model this process in a computer. This difficulty in modeling tasks that seem relatively easy for humans can be seen in other areas of artificial intelligence research, notably research into the task of enabling a computer to recognize faces or creating a two-legged robot that can walk like a human.

The problem with computer modeling of language understanding is that it requires a great deal of knowledge. Humans have extensive knowledge about the way that language is used – knowledge of what individual words mean in various contexts, knowledge of how words can be combined to form clauses and sentences, and knowledge of how sentences can be put together to form some kind of a cohesive story. In addition to this linguistic knowledge, humans also have what has been called *common sense knowledge* – knowledge about people, places and things, the kinds of relationships that exist among objects in the world, and events that might transpire among the various sorts of objects in the world.

Early research into computer models of language understanding – the field of research commonly called *natural language processing (NLP)* – showed that creating a computer program to comprehend something as seemingly simple as a children's story about a birthday party requires an immense inferencing capability [Charniak, 1972]. This research demonstrated the necessity of imposing constraints on the intractable task of full comprehension of human language in order to make the knowledge requirements and inferencing capabilities feasible.

Information extraction is an example of a research area that has benefited from the imposition of a set of constraints on the larger problem of human language comprehension. An information extraction system is intended to extract pieces of information in a text that are relevant to some predefined information need, and then assemble that information into a formal representation. The computer need not fully comprehend everything in the text, both because some portions of most texts are not *relevant* to a given task, and because *shallow processing* – as opposed to deep understanding – is often sufficient for the extraction and assembly of the relevant information. The first section of this chapter will describe the constraints imposed on this area of natural language processing, and highlight the features that make this a rich area for NLP research.

There are many different types of texts from which information could be extracted. News articles are a type of text that is ideally suited to the task of information extraction, since people routinely employ the shallow processing technique of *skimming* newspapers for items of interest, rather than thoroughly reading the entire text of every article. Section 2.2 will explore the application of information extraction systems to the genre of newspaper articles in more detail.

Much of the progress in information extraction has been driven by a series of evaluations – called the Message Understanding Conferences (MUCs) – conducted under the auspices of several United States government agencies. The tasks defined

in these evaluations are of central importance to the field: they constitute the most rigorously defined set of information specifications, information representations and corpora that are widely available to the research community; they therefore provide the framework within which the current work may be most effectively evaluated. Section 2.3 will provide the reader with some background on these tasks.

2.1 Information Extraction: An Application Area of NLP

There are many benefits to conducting research into information extraction. The information extraction task orientation imposes a number of constraints on the more general task of machine understanding of human language – constraints that help to make the information extraction task more tractable than the full comprehension of human language.

Although information extraction is a more constrained task than deep language understanding, many interesting and difficult issues in language processing research must still be addressed by any system that hopes to extract information from a text. Important linguistic issues such as prepositional phrase attachment, processing of conjunctions and appositives, and coreference resolution all figure prominently in this sub-area of NLP research.

One of the advantages of the information extraction task orientation is that it constitutes a well-specified task. The inputs to and outputs from an information extraction system can be defined precisely, which facilitates the evaluation of different systems and approaches. Evaluation of what a system *understands* in a text is much more difficult than evaluation of what a system *extracts* from a text.

A precise specification of the task helps to focus the effort of building an information extraction system. Any system for processing human language is likely to make mistakes – the precise task definition for information extraction helps developers determine the relative importance of different classes of errors.

Many of the problems that arise in developing NLP systems are the result of the need for both broad and deep knowledge of the world. Since the goal of an information extraction system is to extract narrowly defined information from a text, the knowledge requirements of such systems are far less demanding than the requirements for systems that are intended to achieve a broader and deeper understanding of a text. The task is thus well-suited to the use of *shallow processing* models of language.

Narrowly targeted knowledge is easier both to *acquire* and *use* than broad knowledge. A more extensive knowledge base may increase the amount of ambiguity present in the system. By focusing on a specific topic area, or *domain*, the disambiguation problems that pervade all levels of language processing are greatly reduced.

The following sections will elaborate on each of these points.

2.1.1 A Well-Defined Task

In order to develop an information extraction system that will extract certain kinds of information from a collection of documents, the potential users must provide the following resources:

- *Information Specification*: A precise specification of the information requirements for the task, e.g., a listing of the type(s) of information that must be extracted, and any criteria upon which relevancy judgments must be based;
- *Information Representation*: A precise specification of the output representation of the information that is extracted by a system; and

- *Corpus*: A representative collection of documents, some of which contain extractable information (others may be irrelevant), to be used as examples for developing and testing the system.

The information specification is a source of useful constraints – any information not explicitly specified as relevant can be safely ignored by the system. This relevancy criterion often allows an information extraction system to skip over entire sentences and paragraphs in a given text.[1]

The information representation provides a framework for evaluating the system performance, either in isolation or in comparison to the performance of other systems on the same task. This representation is typically in a format that is useful for additional processing by other systems, e.g., automatic entry into a database system, and may or may not be easily readable by humans. The specification of an output format provides additional constraints for language processing – the system need only make those inferences that are necessary to generate the specified output.

A corpus of real texts provides a set of NLP issues on which to focus – the importance of a particular linguistic phenomenon can be given a quantitative assessment based on how frequently it occurs in a corpus. The benefit of this focus is the subject of the next section.

2.1.2 Emphasis on Frequently Occurring Linguistic Phenomena

There is a broad diversity of linguistic phenomena that have been addressed by models and theories in natural language processing. Great progress has been made in some areas, e.g., part-of-speech tagging, while much work remains to be done in other areas, e.g., discourse segmentation.

One benefit of the information extraction task orientation is that it helps to focus effort on linguistic phenomena that are most prevalent in a particular domain, and perhaps in a particular information extraction task. Phenomena that do not arise in a corpus, or that arise relatively infrequently, may warrant less effort, even though some very interesting phenomena may receive little attention.

This corpus-based measure of importance may or may not correspond closely to how *interesting* a particular problem is from the viewpoint of linguistics, psychology or other fields interested in the study of human language use. However, if one is interested in implementing systems that process real texts for specific applications, then an emphasis on commonly occurring problems is a useful guide in deciding how to direct a research effort.

2.1.3 Shallow Processing vs. In-Depth Understanding

As was noted in the opening of this chapter, the task of creating a system that can achieve a deep understanding of a text is very difficult, even for a simple children's story. This is due, in part, to the broad network of possible inferences that is generated with each new sentence.

An information extraction system that analyzes stories about Latin American Terrorism[2] might need knowledge about the names of terrorist organizations and political figures, and perhaps some knowledge about political figures often being the

[1] Some information extraction systems perform a quick assessment of relevancy for each sentence (or paragraph), looking for key words and phrases, and only proceed with a deeper analysis if it appears likely that the text segment is relevant.

[2] See Section A.3 for more information about this domain.

targets of terrorist actions. However, the system may not need to know much about what led to the formation of any particular terrorist organization, or about the political history of any one political figure, or even about the history of the relationship between a particular political figure and a particular terrorist organization. More importantly, issues such as inferring the motivations for particular terrorist actions, or relationships among organizations or individuals – while interesting issues – might be avoided.

2.1.4 Reduction in Ambiguity

The problem of ambiguity pervades all levels of natural language processing. Ambiguities regarding clause boundaries, sentence boundaries, part-of-speech labels and word meanings all complicate sentence analysis. Many common and important sources of ambiguity become simplified when the domain is restricted.

For example, in a corpus of 1000 news articles that focus on business tie-ups or joint ventures, the word "joint" occurs 2004 times; in every instance throughout this corpus it used as an adjective. If the corpus were made up of medical articles that focus on anatomy or of information about carpentry or housing construction, one might expect that "joint" would be used as a noun in the majority of instances.

More importantly, perhaps, out of the 2004 instances of the word "joint", 98% of them are used in contexts that denote some sort of joint business activity.[3] Not only does the corpus constrain the part-of-speech ambiguity for many words, but it also may help to identify certain words that are highly correlated with relevant information in that corpus.

Another example of this phenomenon can be seen in a corpus of 1300 newswire stories on Latin American terrorism. The word "windows" occurs 43 times in this corpus; in every case, the word refers to a window of some sort of physical structure (as opposed to "windows of opportunity"). Furthermore, in all but four of these instances, the word is used in a context that describes a terrorist bombing event – where windows were broken, shattered, etc. – which is likely to be a piece of relevant information in that corpus.

2.2 Extracting Information from News Articles

News articles are particularly well-suited for the task of information extraction. Two of these aspects are discussed below.

2.2.1 Human News Readers as Information Extractors

The goal of reading a news story is usually to acquire some information about some topic of interest to the reader. This information-seeking goal can be contrasted with other goals that people pursue when reading other types of material, goals such as pleasure or spiritual development.[4] Thus the goal of a human news reader is quite close to the goal of a computer information extraction system.

The process of news reading is also compatible with the process of information extraction. A person who reads a news story is often interested in only a fraction of all information in the story – many people routinely "skim" newspapers and other

[3] The other 2% refer to other joint activities such as joint statements or joint military maneuvers.

[4] Of course, one can – and hopefully often does – experience pleasure in seeking (and attaining) information, and there is certainly information to be sought in the pursuit of spiritual development.

sources of news, rather than reading every article in depth. This text-skimming model is often used in the development of information extraction systems.

2.2.2 News Reporters as Information Providers

The goals of most news reporters include satisfying the goals of the intended news readers, one of which is to be able to extract information that is of interest. News articles are thus often written to facilitate information extraction. For example, the first paragraph of a news article typically includes the most relevant information, including the people, places and things – and the important relationship(s) among them – that constitute the focus of the article.[5] A news reader can then determine early on whether the rest of the article is worth reading.

2.3 The Message Understanding Conferences (MUCs)

Much of the recent progress in information extraction has been driven by a series of evaluations, or message understanding conferences (MUCs), conducted under the auspices of several United States government agencies.[6] The tasks defined in these evaluations are of central importance to the field: they constitute the most rigorously defined set of information specifications, information representation formats and corpora that are widely available to the research community;[7] they therefore provide the framework within which the current work may be most effectively evaluated.

A history of the first four MUCs can be found in Appendix A. Brief descriptions of the most recent two MUCs will be provided below, since much of the work described in this dissertation is influenced by these two evaluations.

2.3.1 MUC-5: Joint Ventures

Participants in the Fifth Message Understanding Conference (MUC-5) were provided with four distinct corpora that varied along two dimensions: language – English (E) or Japanese (J) – and domain – Joint Ventures (JV) or Microelectronics (ME) [Onyshkevych *et al.*, 1993]. These corpora were denoted by the acronyms EJV, JJV, EME and JME.

The EJV domain is the focus of the experiments reported in Chapters 7 and 8. This domain focused on business tie-ups.[8] For each tie-up, a system was required to extract information about the organizations involved in the joint venture, the people

[5] Some articles, particularly those written about sporting events or figures, begin with a clever metaphor of some kind, but most articles start with a more straightforward, factual reporting of the key elements of the story.

[6] The MUCs were sponsored by the Advanced Research Projects Agency (ARPA) – formerly known as the Defense Advanced Research Projects Agency (DARPA) – and conducted by the Naval Command, Control and Ocean Surveillance Center, RDT&E Division (NCCOSC/NRaD) – formerly known as the Naval Oceans Systems Center (NOSC).

[7] It should be noted that a research group has to participate in a MUC in order to gain access to these materials, some of which have copyright protection. However, the only requirements for participation in a MUC are that a system accept a text as input and generate a response template as output – there is no requirement of a minimum level of system performance.

[8] A *tie-up* is a relationship among two or more organizations (companies, governments, and/or people) created to achieve some business goal, such as marketing or producing a product often in a new country.

associated with these organizations, the facilities used or owned by the new company, and the products or services provided by the new company.

More details about the EJV domain, the three other domains, and other aspects of MUC-5 can be found in Appendix B.

2.3.2 MUC-6: Corporate Management Changes

The domain chosen for the Sixth Message Understanding Conference ([MUC-6, 1995]) was corporate management changes, e.g., articles about a corporate officer leaving a position in a company or assumes a position in a company (or both). MUC-6 comprised four different, but related, information extraction tasks:

- *Named Entity Recognition*: the identification of proper names referring to people, organizations, geographical locations and dates, as well as references to time, money or percentages.
- *Coreference Resolution*: the establishment of links between phrases in a text that co-refer.
- *Template Element Filling*: the collection of all relevant information concerning each company and person mentioned in a text.
- *Scenario Template Filling*: the establishment of relationship links among all the relevant entities (people, their positions and their corporate affiliations) described in a text.

The MUC-6 Coreference Resolution task will be described in greater detail in Chapter 9. More details on the other tasks can be found in Appendix C.

CHAPTER 3

COREFERENCE RESOLUTION

In its unconstrained form, a solution to the coreference problem is beyond the capabilities of any coreference resolution system. Therefore, several constraints were imposed on the problem throughout the work reported in Chapters 7 and 8.[1]

One of the design goals motivating the development of RESOLVE was to create a coreference resolution system that could be embedded in an information extraction system. Since information extraction systems are designed to extract only the relevant information in a text, the relevancy criterion has been used to constrain the types of phrases that were identified as candidates for coreference resolution by RESOLVE. Examples of relevant and irrelevant phrases are presented in the first section below.

Additional constraints have been imposed in order to make the task more manageable. These include assumptions about the types of noun phrases that are proposed as candidates for coreference resolution and the types of coreference relationships among phrases that would be considered for the task. Constraints arising from these assumptions are described in the second section below.

3.1 Constraints Based on the Information Extraction Task Orientation

One of the benefits of the information extraction task orientation is that it constrains the amount of knowledge necessary to process a text. Rather than requiring full understanding of every phrase in every sentence, phrases that do not contain information that is relevant to the task can be safely ignored, and in some cases entire sentences can be ignored. The phrases that do contain relevant information often do not have to be analyzed as deeply or thoroughly as might otherwise be necessary.

This relevancy constraint simplifies some aspects of the coreference resolution task, in that not all phrases in a text are candidates for coreference resolution. However, the relevant phrases that will be candidates for coreference resolution still represent a broad spectrum of referring expressions: proper names, pronominal references, definite references and even some indefinite references (see Section 3.2.2.3 below).

3.1.1 Relevant Entities

An information extraction task defines a set of entities and possibly a set of relationships among those entities that are relevant to a prespecified information need. Many entities, e.g., companies and governments, were very important to the MUC-5 EJV task, but only if they were somehow involved in a joint venture. Consider the following two sentences, each from a different text in the EJV corpus:

[1] Some of these constraints were relaxed for the work reported in Chapter 9.

```
Maruti Udyog, Ltd., a joint venture between Suzuki Motor
Corp. (7269) and the Indian Government, plans to construct
a second plant in India, in a bid to keep up with increased
demand for its vehicles.

DAIHATSU HAS BEEN AUTHORIZED BY THE INDONESIAN GOVERNMENT TO
PRODUCE 1,000-C.C. GASOLINE AND 2,500-C.C. DIESEL ENGINES IN
THAT COUNTRY UNDER ITS POLICY OF PROMOTING DOMESTIC CAR
ENGINE PRODUCTION.
```

In the first sentence, **the Indian Government** is one of the parents of the joint venture, **Maruti Udyog, Ltd.**, and is therefore a relevant entity according to the the MUC-5 task definition. **THE INDONESIAN GOVERNMENT**, in the second sentence, is not considered a relevant entity within the context of its source text, since authorizing production does not constitute direct involvement in a joint venture.

The information extraction system jointly developed by the University of Massachusetts and Hughes Research Labs for MUC-5 (hereafter referred to as the UMass/-Hughes MUC-5 system) would have identified **the Indian Government** as a relevant entity, using the pattern "**a venture between** *X* **and** *Y*." The system would not have identified **THE INDONESIAN GOVERNMENT** as a relevant entity, since it did not have a pattern such as "**authorized by** *Z*."

Since references to irrelevant entities are not likely to be identified by an information extraction system, such references are not used as candidates for coreference resolution in the experiments reported in Chapters 7 and 8.

3.1.2 Relevant References

Each reference to a relevant entity within a text typically contributes new information about the entity, or its relationships with other entities mentioned elsewhere in the text. Consider the first two sentences from a MUC-5 EJV text:

```
JAPAN AIRLINES CO. (JAL) AND TOYO REAL ESTATE CO., A
SUBSIDIARY OF SANWA BANK, BOUGHT A 122 MILLION U.S. DOLLAR
LUXURY HOTEL TUESDAY, MALAYSIA'S FIRST HOTEL TO BE WHOLLY
OWNED BY JAPANESE.

THE HOTEL, TO BE CALLED HOTEL NIKKO, IS STILL UNDER
CONSTRUCTION AND WAS SOLD BY ONE OF MALAYSIA'S LARGEST
CONGLOMERATES, THE LION GROUP.
```

In this text, the context surrounding the first reference to the hotel – **A 122 MILLION U.S. DOLLAR LUXURY HOTEL** – established the ownership of the hotel (by JAL and Toyo Real Estate), the second reference – **MALAYSIA'S FIRST HOTEL** – provided the location of the hotel (Malaysia), and the third reference – **THE HOTEL, TO BE CALLED HOTEL NIKKO** – included the name of the hotel (Hotel Nikko).

However, some references contribute new information that is irrelevant to an information extraction task. For example, the following sentence occurs later in the same text:

```
JAL, JAPAN'S LARGEST AIRLINE AND ONE OF THE BIGGEST CARRIERS
IN THE WORLD, WILL OPERATE THE HOTEL.
```

The reference to THE HOTEL in this sentence contributes no information relevant to the MUC-5 EJV task – extracting the operator of a facility was not part of the task specifications.

An information extraction system may ignore such references, or if it does identify irrelevant references, it may not be able to extract much information about them. Since the knowledge about irrelevant references to relevant entities is likely to be quite limited, these references are excluded from the data used in the experiments reported in Chapters 7 and 8.

3.2 Additional Constraints

The task orientation of an information extraction system imposes a set of constraints relating to the relevancy of references throughout a text. However, a number of other decisions must be made in defining the coreference task to make evaluation feasible.

3.2.1 Noun Phrases

Noun phrases, or NPs, are candidates for coreference resolution; modifying nouns are not considered. For example, the sub-phrase FORD MOTOR CO. within the phrase FORD MOTOR CO.'S EUROPEAN UNIT was not considered as a candidate for coreference resolution.

This restriction on modifying nouns was based on an assumption concerning the type of sentence analysis required for information extraction. The component that analyzed noun phrases within the UMass/Hughes MUC-5 system was only concerned with extending simple noun phrases to include relative clauses and prepositional phrases; no attempt was made to identify nested sub-phrases that referred to distinct entities. Since RESOLVE was intended to work with an information extraction system such as the UMass/Hughes MUC-5 system, it was not given such nested sub-phrases during training or testing.

Fortunately, nested sub-phrases that refer to distinct entities are found in only five references out of the 628 that formed the basis for the experiments reported in Chapters 7 and 8.

The following sections will elaborate on the definition of a noun phrase that was used in the work reported in this dissertation.

3.2.1.1 Simple NPs

A *simple noun phrase (simple NP)* is a sequence of words, possibly starting with an article, containing any number of modifying words (adjectives, adverbs or nouns) and ending with a head-noun. In BNF-notation, this definition is:

$$
\begin{array}{lcl}
\text{SimpleNP} & = & [\textit{article}][\text{Noun-Modifier}] * \textit{head-noun} \\
\text{Noun-Modifier} & = & \textit{adjective} \\
\text{Noun-Modifier} & = & \textit{adverb} \\
\text{Noun-Modifier} & = & \textit{noun}
\end{array}
$$

Examples of simple NP's include

- IT

- TOYO REAL ESTATE
- A LOCAL COMPANY
- THE THIRD LARGEST BRAZILIAN LIME MAKER

3.2.1.2 Complex NPs

Many entity references are made up of complex noun phrases, e.g., attached prepositional phrases, parenthetical phrases, relative clauses or appositive constructions. The noun phrase analysis module of the UMass/Hughes MUC-5 system was used to extend simple noun phrases to include these more complicated variations. Such complex noun phrases were therefore used in the evaluation of RESOLVE reported in this dissertation.

The following formula provides BNF definition of a noun phrase that includes both the simple forms and more complicated forms:

$$
\begin{array}{lcl}
\textit{SimpleNP} & = & [< \textit{article} >][< \textit{noun-modifier} >]* < \textit{head-noun} > \\
\textit{NP} & = & \textit{SimpleNP} \\
\textit{NP} & = & \textit{NP} < \textit{preposition} > \text{emphNP} \\
\textit{NP} & = & \textit{NP}(\textit{NP}) \\
\textit{NP} & = & \textit{NP} < \textit{past participle verb phrase} > \text{emphNP} \\
\textit{NP} & = & \textit{NP}, \textit{NP}
\end{array}
$$

Examples of noun phrases with different levels of complexity include:

- Prepositional Phrase: YAKULT HONSHA CO. OF JAPAN has two constituent simple NPs: YAKULT HONSHA CO. and JAPAN.
- Parenthetical Phrase: JAPAN AIRLINES CO. (JAL)
- Relative Clause: THE JOINT VENTURE, CALLED P.T. JAYA FUJI LEASING PRATAMA has two constituent simple NPs: THE JOINT VENTURE and P.T. JAYA FUJI LEASING PRATAMA.
- Appositive: THE NEW FIRM, P.T. FUJI DHARMA ELECTRIC has two constituent simple NPs: THE NEW FIRM and P.T. FUJI DHARMA ELECTRIC.
- Combination: SUMITOMO, JAPAN'S THIRD LARGEST STEELMAKER BASED IN OSAKA, WESTERN JAPAN has three constituent simple NPs: SUMITOMO, JAPAN'S THIRD LARGEST STEELMAKER and OSAKA, WESTERN JAPAN.[2]

3.2.2 Types of Coreference

There are many ways that a phrase can refer to something already mentioned in a text. Carter [1987] provides an extensive list of the ways that a phrase may refer to something either implicitly or explicitly referenced earlier in that text.

This section will focus on the type types of coreference that have been found among relevant references in the MUC-5 EJV corpus.

[2]Location descriptions that include commas were not separated.

3.2.2.1 Identity Coreference

The simplest, and most common, type of coreference is *identity* coreference, i.e., two phrases refer to identical entities.[3] Consider the following two sentences from a text.

```
FAMILYMART CO. OF SEIBU SAISON GROUP WILL OPEN A CONVENIENCE
STORE IN TAIPEI FRIDAY IN A JOINT VENTURE WITH TAIWAN'S
LARGEST CAR DEALER, THE COMPANY SAID WEDNESDAY.

THE JOINT VENTURE, TAIWAN FAMILYMART CO., IS CAPITALIZED AT
100 MILLION NEW TAIWAN DOLLARS, HELD 51 PCT BY CHINESE
AUTOMOBILE CO., 40 PCT BY FAMILYMART AND 9 PCT BY C. ITOH
AND CO., A JAPANESE TRADING HOUSE.
```

There are three sets of coreferent phrases that provide examples of identity coreference between relevant phrases:

- FAMILYMART CO. and FAMILYMART
- TAIWAN'S LARGEST CAR DEALER and CHINESE AUTOMOBILE CO.
- A JOINT VENTURE and THE JOINT VENTURE, TAIWAN FAMILYMART CO..

There also exists a relationship of identity coreference between the two references to Taiwan: TAIWAN'S LARGEST CAR DEALER and 100 MILLION NEW TAIWAN DOLLARS. However, this relationship will not be considered further, both because Taiwan is not an *organization* in the sense used for the MUC-5 EJV corpus, and because in each case, TAIWAN constitutes a nested sub-phrase, a construct which is excluded (see Section 3.2.1).

3.2.2.2 Subset-Superset Coreference

Articles about joint ventures sometimes refer to a group (or set) of joint venture parents and then later contain more specific references to the individuals making up that group. This sort of coreference relationship might be described as a *superset/subset* relationship. In other situations, individual entities might be mentioned first, and then later references may group some of those entities together; this relationship might be described as a *subset/superset* relationship.

Two sentences from the same text illustrate both of these types of relationships:

```
MITSUI PETROCHEMICAL INDUSTRIES LTD. AND MITSUI AND CO. WILL
SET UP A JOINT VENTURE IN INDONESIA WITH TWO LOCAL FIRMS TO
PRODUCE PURIFIED TEREPHTHALIC ACID (PTA) MATERIAL FOR
POLYESTER FIBER, THE TWO JAPANESE FIRMS ANNOUNCED MONDAY.

THE JOINT COMPANY, PTA INDONESIA, WILL BE CAPITALIZED AT 50
MILLION DOLLARS, OF WHICH 50 PERCENT WILL BE PROVIDED BY
MITSUI PETROCHEMICAL, 20 PERCENT BY MITSUI AND CO., A
JAPANESE TRADING GIANT, 20 PERCENT BY THE INDONESIAN
STATE-OWNED PETROLEUM AND NATURAL GAS MINING ENTERPRISE,
PERTAMINA, AND 10 PERCENT BY PERTAMINA'S SALES ARM, HUMPUSS.
```

[3] Identity coreference is also called *cospecification* by Sidner [1979], to emphasize the fact that two coreferent phrases do not so much refer to each other as much as they co-specify the same entity stored in the system's database.

The phrase THE TWO JAPANESE FIRMS is coreferent with both MITSUI PETROCHEMICAL INDUSTRIES LTD. and MITSUI AND CO., and since the superset reference comes after the individual (subset) references, this is an example of the *subset/superset* coreference relationship.

The two phrases THE INDONESIAN STATE-OWNED PETROLEUM AND NATURAL GAS MINING ENTERPRISE, PERTAMINA and PERTAMINA'S SALES ARM, HUMPUSS are both coreferent with the earlier phrase TWO LOCAL FIRMS. This is an example of the *superset/subset* coreference relationship.

3.2.2.3 General/Specific Coreference

Sometimes, a specific entity will be referenced and then a subsequent reference will be to a more general class, of which the specific entity is a member. The second reference is often an indefinite reference, and thus may not be considered an anaphoric reference, but must be linked somehow with the earlier reference, since it adds important information about that entity.

As an example, consider the following sentences:

> FAMILYMART CO. OF SEIBU SAISON GROUP WILL OPEN A CONVENIENCE STORE IN TAIPEI FRIDAY IN A JOINT VENTURE WITH TAIWAN'S LARGEST CAR DEALER, THE COMPANY SAID WEDNESDAY.
>
> THIS WILL BE THE FIRST OVERSEAS STORE TO BE RUN BY A JAPANESE CONVENIENCE CHAIN STORE OPERATOR.

The indefinite reference A JAPANESE CONVENIENCE STORE OPERATOR is a rather general reference to a class of entities, i.e., Japanese convenience store operators. However, since an information extraction system working in the EJV domain needs to extract nationality information about organizations, a coreference resolution module needs to be able to link this indefinite reference with the earlier reference to FAMILYMART CO. OF SEIBU SAISON GROUP. This is an example of the *specific/general* coreference relationship, since the more general reference occurs later than the more specific one.

Indefinite references followed by more definite references are usually considered cases of *identity* coreference relationships, as was the case with A JOINT VENTURE and THE JOINT VENTURE, TAIWAN FAMILYMART CO. described in an earlier section.

3.3 Transitive Closure of the Coreference Relation

Coreference is a transitive relation among phrases in a text, and identity coreference is a symmetric relation. For example, if we know that phrase A is coreferent with phrase B, and that phrase B is coreferent with phrase C, then we can conclude that phrase A is coreferent with phrase C.

More concretely, consider the following three sentences:

> FAMILYMART CO. OF SEIBU SAISON GROUP WILL OPEN A CONVENIENCE STORE IN TAIPEI FRIDAY IN A JOINT VENTURE WITH TAIWAN'S LARGEST CAR DEALER, THE COMPANY SAID WEDNESDAY.
>
> THE JOINT VENTURE, TAIWAN FAMILYMART CO., IS CAPITALIZED AT 100 MILLION NEW TAIWAN DOLLARS, HELD 51 PCT BY CHINESE AUTOMOBILE CO., 40 PCT BY FAMILYMART AND 9 PCT BY C. ITOH AND CO., A JAPANESE TRADING HOUSE.

TAIWAN FAMILYMART PLANS TO OPEN SEVEN MORE STORES IN TAIPEI IN DECEMBER, AND HOPES TO OPEN 200 STORES THROUGHOUT TAIWAN IN THREE YEARS.

If we know that A JOINT VENTURE (in the first sentence) is coreferent with THE JOINT VENTURE, TAIWAN FAMILYMART CO. (in the second sentence), which, in turn, is coreferent with TAIWAN FAMILYMART (in the third sentence), then we can conclude that A JOINT VENTURE is also coreferent with TAIWAN FAMILYMART.

The transitive nature of the coreference relation has important ramifications for a system that classifies phrases as coreferent or not coreferent: the links between the first and second references to the Taiwan Familymart and between the second and third references to the company are easier to establish than the link between the first and third reference. This is because the first and second references both contain the sub-phrase JOINT VENTURE and the second and third references both contain the sub-phrase TAIWAN FAMILYMART. The first and third references have nothing in common; in fact, if the second sentence were removed, a human reader would likely have some difficulty establishing a coreference link between these two references.

If a system were to misclassify the first and third references to Taiwan Familymart as not coreferent, this information could still be recovered by virtue of the transitive closure of the other two links (between the first and second, and second and third references). This potential for capturing all the important information without having correctly classified *every* pair of phrases has important implications for evaluating performance on the coreference classification task. These implications will be discussed in greater detail in Section 6.3. For now, it is sufficient to note that a simple measure of classification accuracy may not be adequate for performance evaluation.

3.4 Implications of Coreference Resolution for Information Extraction

Accurate coreference resolution is very important for effective processing of non-trivial natural language texts, whether the text processor is a computer or a human. A language processing system that is too *liberal* in resolving references, i.e., errs in the direction of incorrectly classifying phrases with distinct referents as coreferent, will fail to make distinctions among separate entities. In the extreme case, such a system might resolve all references to a single entity. A text that mentions only one thing, without reference to anything else, is presumably quite rare.

In contrast, a system that employs a *conservative* coreference resolution strategy, i.e., errs in the direction of incorrectly classifying coreferent phrases as not coreferent, may make too many distinctions among references to the same entity. The extreme example of this would be for a system to presume that every reference has a distinct referent. A text that never mentions anything more than once is another rarity.

The effect that a liberal or conservative bias in coreference classification has on the performance of a larger NLP application is an open question. Some ideas about the effect of this bias upon the performance of an information extraction task will be discussed in Section 10.3.5.

3.5 Other Work on Coreference Resolution

Coreference resolution has long been recognized as an important and difficult problem by researchers in Linguistics, Philosophy, Psychology and Computer Science. The work described in this dissertation is oriented toward a solution to coreference

resolution that can be implemented by a computer program, and this computational orientation will provide the focus for the discussion of related work in this section.[4]

3.5.1 Early Research

Two early natural language processing systems highlighted many of the issues that arise in the development of a computer program to resolve references to previously mentioned entities. Charniak [1972] created a system for understanding children's stories, a genre that proved far more complex than might be imagined. Winograd [1972] built a system for interacting with a human in natural language about an imaginary *micro-world* consisting of blocks on a table that could be moved about by a robot arm.

3.5.1.1 Deep Semantic Processing (DSP)

One of the earliest computer programs to confront the problem of coreference resolution was a system created by Charniak to understand simple children's stories [Charniak, 1972]. An example posited by Charniak (page 7) is the problem of resolving the pronoun `it` in the last sentence of the following short story:

> `Today was Jack's birthday. Penny and Janet went to the store. They were going to get presents. Janet decided to get a top. "Don't do that" said Penny. "Jack has a top. He will make you take it back."`

In order for a reader to determine that `it` (sentence 7) refers to the top that Janet intended to buy for Jack (sentence 4), and not the top that Jack already has (sentence 6), a number of deductions must to be made (page 63):

- The presents (sentence 3) are intended for Jack.
- The top (sentence 4) would be Janet's present for Jack.
- If Jack already has a top (sentence 6), then he might not want another top.
- If Jack does not want another top, then he might make Penny return the top.

Two aspects of Charniak's work are important to note here. One is that an enormous amount of *common-sense knowledge* is required to understand even a simple children's story; in the example above, a reader needs to know about birthday parties (children typically bring presents to birthday parties), stores (a place to buy presents), and tops (children typically have at most one top, unlike, say, model airplanes or dolls). The second is that a large number of *inferences* may be required for using common-sense knowledge to understand a simple story (four of which are listed above).

Much of the work on coreference resolution that has been done since Charniak's thesis has been concerned with constraining the amount of common-sense knowledge required by a natural language processing system or controlling the amount of inferencing (or searching) that is done by a system.[5] The other theories, and implementations of those theories, that are discussed below represent different strategies to accomplish these goals.

[4] See Sidner [1979], Hirst [1981] or Carter [1987] for excellent accounts of a broad range of research into coreference resolution.

[5] These themes – minimizing the knowledge requirements and controlling search – are common themes throughout much of the research in Artificial Intelligence.

3.5.1.2 The Blocks World

One way of limiting the common-sense knowledge required for a natural language understanding system is to impose restrictions on the "world" that provides the context in which a discourse takes place. Winograd [1972] created an imaginary *micro-world* inhabited only by a robot (with a hand for manipulating objects and an eye for seeing), a table, a box and a set of eight toy blocks of varying shapes, sizes, colors and locations. All discourse takes place in a dialog between a user – who can issue commands and ask questions – and the robot – who can carry out commands, answer questions and ask questions itself (for clarification purposes).

The micro-world simplifies many of the problems in coreference resolution:

- The pronouns `I` and `you` are always resolved to the user or the robot, depending on which participant has typed the sentence in which the pronoun(s) occur;
- The pronoun `it` can only refer to one of the 10 objects in the world (the table, the box or one of the blocks).

Winograd specifies a set of procedures for determining the referents for each of three different classes of reference:

1. `it` or `they`
2. numbers such as `one` or `two`, or more complex constructions such as `at least three`.
3. definite descriptions such as `the two red blocks`

These procedures take into account many factors that are used in subsequent research, such as recency, surface syntactic structure, and the presence and form of determiner. They encode preferences such as preferring the focus of a clause over other elements of the clause and ranking clause elements in the order of subject, direct object, prepositional phrase and secondary clauses.

The evaluation of Winograd's system consists mostly of an extended dialog presented in the first chapter. For example, in the first sentence the user issues the following command to the robot:

```
Pick up a big red block.
```

Since the system has complete knowledge about the shape, size, color and location of every block, it is able to carry out this action by picking up the largest red block on the table, labeled B5.[6] In the third sentence of the sample dialog, the user issues another command to the robot:

```
Find a block which is taller than the one you are holding
and put it into the box.
```

The system is able to

[6]The robot first moves a green block which is on top of the largest red block. The ***largest*** red block is selected because there are only three red blocks, and the definition of ***big*** used by the system is to select an object that fits other elements of the description – in this case ***red blocks*** – such that "the number of objects fitting the description and smaller than the one being described is more than the number of suitable objects bigger than it is." [page 129]

1. Determine the referent of **one**, in the phrase **the one you are holding**, as B5, the block currently in the robot hand.

2. Select a referent for **a block**, in the phrase **a block which is taller than the one you are holding**. Since B5 is **the one you are holding**, and B10 is the only block taller than B5, B10 is selected.

3. Determine the referent of **it**, which is determined to be **a block which is taller than the one you are holding**, or B10, based on a search order that checks previous clauses in the same sentence before checking a previous sentence, and a preference for resolving **it** with a subject (**a block**) over a noun group in a secondary clause (**the one you are holding**).

Winograd raised a number of issues that are still considered open problems for research into coreference resolution (and many other aspects of natural language processing), and his sample dialog is a very impressive display of natural language understanding by a computer. Unfortunately, a system that is intended to process *real-world* texts does not enjoy the luxury of knowing everything that is potentially relevant to all of the objects in its world. However, it should be noted that until fairly recently [Connolly *et al.*, 1994, Aone and Bennett, 1995, McCarthy and Lehnert, 1995], most of the theories and systems developed for coreference resolution were provided with *all* of the knowledge they required for finding correct antecedents for all of the anaphors upon which they were tested.

3.5.2 Focusing Theory

A coreference resolution system must determine whether one phrase refers to a preceding phrase in a text, often by searching through a list of preceding phrases with which a new phrase may co-refer. One way of reducing the inference load of a coreference resolution system is to limit the set of preceding phrases that are considered as possible antecedents to a new phrase.

Focusing Theory constrains the search for an antecedent by defining a set of entities that constitute the *current focus* at any given point in a text; when searching for antecedents, the entities in the current focus are considered as possible antecedents before other entities are considered. The theory also defines the ways that a writer may signal a *shift of focus* that marks a different set of entities as the current focus.[7]

3.5.2.1 Focus Spaces

Grosz [1977] defined a representation for tracking the focus of a discourse: an *explicit focus space* that includes all of the entities explicitly referenced in a discourse segment[8], and an *implicit focus space* that includes all of the entities that are associated with entities in an explicit focus space. At any given time, there is one *active* focus space (explicit and implicit) representing the current discourse segment and a set of *open* focus spaces representing previous discourse segments to which the focus may shift. The entities in a focus space were encoded as an elaborate semantic network.

The domain to which this representation of focus spaces was applied was the interpretation of a task-oriented dialog, the task being the assembly of components

[7] The sets of entities that constitute different foci need not be disjoint, i.e., an entity can remain in focus even after the focus shifts.

[8] A discourse segment was referred to as "the context of an utterance" in this work.

such as an air compressor. The explicit focus contained items explicitly referenced in the dialog; the implicit focus contained related items, e.g., subparts of an assembly or subtasks associated with the main task. The active focus space represents the current step of an assembly procedure; the open focus spaces represent any previous steps that were not completed.

One of the primary contributions of this work was in constraining the search procedure for resolving definite noun phrases. When a definite noun phrase was encountered, the search procedure attempted to make a "focused match" between a network structure representing the context in which the definite noun phrase was used and the network structure in the active focus spaces. Another contribution of this research is that it contains a description for how to detect shifts of focus.

One problem not addressed by this work was the resolution of pronominal references. Another problem was the detection of focus shifts in discourses outside the area of task-oriented dialogs: the task of assembling components is fairly well-defined and subtasks can be neatly delineated, making the recognition of which focus spaces are still *open* simpler than it might be in other domains; since keeping a small number of focus spaces open is necessary to effectively constrain the search for antecedents, detecting a shift in focus is an important problem that would need to be addressed in applying this research to other domains.

3.5.2.2 The Local Focusing Framework

Sidner [1979] extended the idea of focusing to include pronouns (Grosz's theory only accounted for non-pronominal noun phrases) and an elaboration on recognizing shifts in a discourse. Under Sidner's theory of local focusing, the resolution of definite pragmatic anaphora[9] is tightly coupled with tracking the focus of a discourse. Definite anaphora resolution affects the current discourse focus, signaling either a retention of the current focus or a shift to a new focus; the current focus also influences the resolution of definite anaphora by constraining the search for antecedents.

Sidner proposed a set of data structures for tracking local focus – including an *actor focus*, a *current focus* and several lists containing other elements mentioned in the discourse – an elaborate set of rules for proposing referents for definite anaphors and another detailed set of rules for determining the new current focus. For each sentence, the anaphors in a sentence are resolved, and then the data structures are updated, resulting either in the retention of the current focus or a shift of focus to another element of the discourse.

The phenomena that Sidner is modeling are quite complex: the use of definite anaphora in English discourse. Unfortunately, the model she proposes is also quite complex – in addition to maintaining a large number of data structures for tracking local focus and candidates for anaphor resolution, there are a number of different algorithms for applying this framework in very specific contexts – e.g., different algorithms for "Third Person Pronoun in Agent Position," "Third Person Pronoun in non-Agent Position" and "Third Person Pronoun Personal Possessive Pronouns" – and many of these algorithms are rather long and complicated.[10]

Another shortcoming of Sidner's work is that it is intended primarily for the processing of simple sentences, e.g., those of the form:

subject verb-phrase direct-object [*indirect-object*] [*prepositional-phrase*]*.

[9] Pronouns and other definite noun phrases that refer to antecedents, as opposed to pronouns that occur in a context such as "It is raining."

[10] See Sidner [1979], Appendix B, for descriptions of these algorithms.

While more complex sentences are included in the set of sentences to which she applied her algorithms, they appear to be segmented into simpler constituents based on intuition, i.e., no algorithm is provided for segmenting complex sentences into the simpler constituents for which her algorithm is intended.

The complexity of the Sidner's algorithm and the need to account for complex sentences have both been addressed by subsequent research, which is described below.

3.5.2.3 The Centering Framework

Grosz, Joshi and Weinstein (GJW) [1983] proposed a theoretical framework, called *centering*[11], as an alternative to Sidner's local focusing framework. Centering theory was an attempt to simplify both Sidner's data structures and her algorithm. Centering requires only two data structures for tracking the local focus of a sentence or utterance:

- A *backward-looking center*, $C_b(U_i)$, which is an element of the previous utterance (U_{i-1}) that constitutes the focus of the current utterance (U_i); this element is sometimes referred to simply as the *center* of the utterance.
- A set of *forward-looking centers*, $C_f(S_i)$, which are the discourse elements of the current utterance, ranked in order of their predicted likelihood to become $C_b(S_{i+1})$;[12] the highest ranked element of this list is sometimes called the *preferred center* or $C_p(U_i)$.

Another motivation behind centering theory was the need to address the use of definite noun phrases by a natural language generation (NLG) system; most previous approaches to definite anaphora had addressed only natural language understanding. The orientation toward NLG, combined with the intention to simplify the algorithms that Sidner proposed for tracking local focus, leads to the following single rule used in centering theory:

> If the C_b of the current utterance is the same as the C_b of the previous utterance, a pronoun should be used.

A corollary to this rule is that if any element of an utterance is a pronoun, then the $C_b(U_i)$ should be a pronoun (there might be other pronominal references as well).

In addition to this rule concerning pronoun usage, later formulations of centering theory [Grosz *et al.*, 1986, Grosz *et al.*, 1995] also defined three types of transition relations for *center movement* between pairs of adjacent utterances:

1. *Continuation* of the center (C_b) from one utterance not only to the next, but also to subsequent utterances $(C_b(U_i) = C_b(U_{i+1}) = C_p(U_{i+1})$.
2. *Retention* of the center from one utterance to the next, but probably not to subsequent utterances $(C_b(U_i) = C_b(U_{i+1}) \neq C_f(U_{i+1})$.
3. *Shifting* the center, if it is neither retained nor continued $(C_b(U_i) \neq C_b(U_{i+1}))$.

In addition to defining this set of transition relations, the later formulations also define a set of preferences that constrain the use of pronouns in a discourse:

> Sequences of continuation are preferred over sequences of retaining; and sequences of retaining are to be preferred over sequences of shifting.

[11] A more recent, and thorough, exposition on the central features of centering theory can be found in GJW [1995]

[12] The order is defined in terms of surface syntactic structure: "subject, object, object2, followed by other sub-categorized functions and finally adjuncts." [Brennan *et al.*, 1987]

Table 3.1 Extended Transition State for Centering Theory

	$C_b(U_i) = C_b(U_{i-1})$	$C_b(U_i) \neq C_b(U_{i-1})$
$C_p(U_i) = C_p(U_{i-1})$	CONTINUING	**SHIFTING-1**
$C_p(U_i) \neq C_p(U_{i-1})$	RETAINING	SHIFTING

3.5.2.4 A Centering Approach to Pronoun Resolution

Centering theory was proposed, in part, to enable a natural language generation system to decide upon the appropriate use of pronouns in *generating* a discourse. Using centering theory for interpreting pronouns in a discourse was a topic explored in subsequent research.

Brennan, Friedman and Pollack (BFP) [1987] extended centering theory by defining a fourth transition relation, *shifting-1*, to distinguish cases where the new center of an utterance (after a shift) is likely to continue as the center of the subsequent utterances from cases where the new center of an utterance is unlikely to continue as the center of an utterance. Table 3.1 (from Figure 3 in BFP [1987]) illustrates the four transition relations based on the backward center (C_b) and preferred forward center (C_p) of a pair of adjacent utterances:

This new transition is included in the ranking of preferences for center movement, inserted between retaining and the original shifting transition defined in earlier work on centering ($\succ$ should be read as "is preferred over"):

$$\textit{continuing} \succ \textit{retaining} \succ \textbf{\textit{shifting-1}} \succ \textit{shifting}$$

They then define an algorithm incorporating these preferences in order to resolve pronominal references, wherein

1. Each pronominal reference in an utterance (U_i) is paired up with each phrase in the previous utterance (from $C_f(U_{i-1})$ with which it agrees [in number and gender]; these expanded references are then combined with the other phrases in U_i to form all possible pairings of backward centers and forward centers $< C_b, C_f >$ for the current utterance, i.e., for each reference in an utterance $R_j(U_i) \in R(U_i)$, the pair $< R_j(U_i), R(U_i) - R_j(U_i) >$ is created.

2. Filter the $< C_b, C_f >$ pairs according to the constraints of centering theory and other constraints such as contraindexing.[13]

3. Rank the pairs according to the transition preferences defined by centering theory.

The centering approach to pronoun resolution is certainly simpler than Sidner's local focusing approach to the same task. However its intended task – pronoun resolution – represents only a portion of the phenomena for which Sidner's approach was designed. Furthermore, the evaluation performed to compare the two approaches – analyzing the behavior of the two sets of algorithms on two simple discourses –

[13]For utterances such as "He invited him to dinner," contraindexing prevents "He" and "him" from being resolved to the same reference from a previous utterance, reflecting the intuition that "he" and "him" refer to distinct people.

was not very rigorous, leaving the issue of which algorithm is "better" as an open question.

Centering theory is based on largely syntactic structure, e.g., the ranking of elements on the C_f list is entirely dependent on the syntactic role played by each element in the utterance. Centering theory is intended only to propose potential referents for a pronoun, not to select *one* of those referents as the antecedent for the pronoun. The need for semantic and pragmatic knowledge for the resolution of pronouns is acknowledged; however the scope of such knowledge is not addressed.

3.5.2.5 The Shallow Processing Anaphor Resolver (SPAR)

Assessing the semantic and pragmatic knowledge required by an anaphor resolution system was one of the motivations behind Carter's research [1987]. The Shallow Processing Anaphor Resolver (SPAR) was developed to incorporate and extend both Sidner's [1979] algorithms for tracking local focus and resolving definite anaphora and Wilk's work on common sense inference (CSI) using *preference semantics* [Wilks, 1975]. SPAR was motivated by the *shallow processing hypothesis*:

> In this approach, linguistic knowledge is exploited as fully as possible, while knowledge of the world, which is notoriously difficult to represent and process adequately, is present only in limited quantities and is invoked only when absolutely necessary. [Page 13]

SPAR uses Boguraev's [1979] sentence analyzer to identify references in individual sentences that are candidates for resolution and to provide the sentence-level linguistic knowledge used during reference resolution. For the candidates (anaphors) in each sentence, SPAR uses a sequence of knowledge sources for resolving them:

1. Word sense information is applied to constrain possible referents.
2. An extended version of Sidner's anaphor resolution rules is used to infer additional constraints on possible referents.
3. Syntactic rules, such as *c-command*, are invoked to eliminate some of the possible referents.
4. A CSI component is used to make inferences about the remaining referents, with the goal of further constraining the set of possible referents.
5. If more than one possible referent remains for any anaphor, a set of special-purpose heuristics is invoked to select one referent.

SPAR's output is a paraphrase of each sentence in a text, with the definite anaphors being replaced by uniquely identifying references (e.g., names of people).

Carter's research represents an important contribution toward assessing the types of knowledge, and the interaction among those different types of knowledge, that are required for resolving anaphors. It also includes one of the earliest attempts at a quantitative analysis of the performance of a system for coreference resolution: SPAR was found to resolve 226 of 242 pronouns (93%) correctly and to resolve 66 of 80 non-pronominal anaphors (82%) correctly. This early attempt at evaluation of a coreference resolution system demonstrates the effectiveness of SPAR on the set of texts to which it was applied.

However, the evaluation of the system suffers from one of the same problems that has affected evaluations of other systems: the texts on which SPAR was tested were constructed specifically for the purpose of testing the system (or for testing earlier systems), they were not *real-world* texts representing naturally occurring discourse. SPAR was tested on a set of 40 texts written by Carter to develop the system, and then on a set of 23 texts written by other people not directly associated with SPAR – some of the texts in this second set were edited so that they would "fall within the analyser's grammatical coverage". Most of the texts were composed of short, simple sentences – the longest sentence contained 13 words.

3.5.2.6 A Focusing Framework for Complex Sentences

One of the shortcomings of most of the work done on focusing or centering is that the algorithms and data structures are constructed to handle short, simple sentences, for example:

subject verb-phrase direct-object [*indirect-object*] [*prepositional-phrase*]*

The longest sentence found in Brennan *et al.* [1987] is 9 words long; the longest sentence in Carter [1987] was composed of 13 words. Moreover, most of the sentences used to evaluate coreference resolution algorithms have been specially constructed in order to test certain aspects of the algorithms.[14]

Suri [1993] defines a two-part Semantically-Slanted Discourse (SSD) Methodology for determining how to extend a framework, e.g., local focusing or centering theory, to process different types of complex sentences. The first part of the methodology involves generating simple discourses each composed of a sequence of sentences, one of which exhibits the specific type of complexity to be investigated, e.g., sentences of the form "SX because SY", where SX and SY each consist of a simple clause; the referent of every noun phrase is fully determined by *semantic factors* alone. The references in sequence of sentences are varied, e.g., substituting pronouns for more fully-specified noun phrases. Native speakers then pass judgment on the appropriateness or awkwardness of each variation, and these judgments are used to guide the extension of a framework to account for the specific type of sentential complexity.

Suri is sensitive to the potential shortcomings of basing extensions to a framework on a set of specially constructed sentences, which reflect the biases (intentional and unintentional) of the generator, and which may or may not be representative of sentences in real-world texts. Therefore, once a framework has been extended, the second part of the SSD Methodology involves a corpus analysis of the effect of the extension.

The SSD Methodology is an important contribution toward solving the problem of application of theoretical frameworks to real-world texts. However, as Suri notes, much more work needs to be done, since the methodology has only been applied to one form of complexity ("SX because SY"), and even that sentence form was constrained to situations in which a pronoun appears in the subject position of SX. Also, a framework for conducting a corpus analysis has yet to be constructed.

3.5.2.7 A Focusing Extension for Embedded Sentences

Another form of complex sentence has been investigated by Azzam [1996]: embedded sentences, which are broadly defined as sentences that concern more than

[14] Walker [1989] evaluated the expected behavior of both the centering algorithm and Hobbs' [1976] pronoun resolution algorithm on a series of real world texts, but the algorithms were hand-simulated.

one fact or *elementary event*, e.g., a sentence that includes both the act of saying something and the thing that is said:

> **Three of the world's leading advertising groups, Agence Havas S.A. of Frane, Young & Rubicam of the U.S. and Dentsu Inc. of Japan, said they are forming a global advertising joint venture.**

Azzam extends Sidner's focusing approach to resolve pronominal references – with both intrasentential antecedents (as above) and intersentential antecedents – in the context of such embedded sentences. These extensions are tested on a corpus of 120 Reuters news articles in the financial domain, both by using a partial implementation of these extensions in conjunction with a sentence analyzer, which achieved a "success rate" of 70% , and by a hand-simulation which achieved a success rate of 95%.[15]

Azzam's approach appears to be well-suited to coreference resolution for information extraction. Unfortunately, there remain obstacles to using her approach in an implemented system. One problem is that the algorithm, as described in Azzam's ACL conference paper [1996], only handles pronoun resolution;[16] as noted above, pronoun resolution plays a relatively minor role in coreference resolution for some domains. Another obstacle to using this algorithm is that the full details of the algorithm are contained in Azzam [1995], which is written in French.[17] However, Azzam's algorithm is a promising approach that may one day prove very effective at coreference resolution within an information extraction system.

[15] See Chapter 6 for a discussion about measurements of accuracy in evaluating coreference resolution performance.

[16] Although the algorithm is currently being extended to handle other types of anaphora (Saliha Azzam, personal communication).

[17] A translation to English is planned, but completion is expected to take quite a while (Saliha Azzam, personal communication).

CHAPTER 4

A TRAINABLE APPROACH

One of the advantages to using a machine learning algorithm is that such algorithms are designed to find regularities in the data. As noted in Section 3.5, much of the previous work on coreference resolution assumed complete knowledge, at least for the sample sentences to which a given approach was applied; and the sample sentences were often constructed specifically for testing a given approach. RESOLVE is unlikely to have complete knowledge of any particular domain, and its input will come from real-world, naturally occurring texts rather than specially constructed texts. However, since corpora to which information extraction systems are typically applied tend to be narrowly constrained, there is likely to be regularity in patterns of coreferring within a corpora – patterns that may be identified by the machine learning algorithm.[1]

In order for any machine learning algorithm to create a classifier for coreference resolution automatically, two primary requirements must be met. First, a representation of the problem must be defined. This representation determines what constitutes an example or *instance* of the problem; the particular aspects or *features* of each instance of the problem that will presented to the learning algorithm; and the *classes* that might be returned by the classifier as a solution to each instance to the problem. Second, an algorithm must be selected – or invented – that will permit a computer to learn how to solve the problem, based on an examination of previous examples of the problem.

The first section of this chapter will describe the problem representation and the algorithm used for coreference resolution. The remainder of the chapter will be devoted to a discussion of other applications of machine learning techniques to problems in natural language processing.

4.1 A Trainable Framework for Coreference Resolution

Several possible frameworks were considered for RESOLVE. This section will describe some of the decisions that were made in establishing the current framework.

4.1.1 Problem Representation

One of the most challenging issues in applying machine learning techniques to coreference resolution – and many other difficult problems – is to decide what constitutes an instance of the problem. The coreference problem is to determine when a phrase refers to something already mentioned in a text. There are several ways of representing this problem, four of which will be described below (readers interested only in the representation actually selected for this problem can skip to Section 4.1.1.4.)

[1] Chapter 9 examines one such pattern in detail.

Examples will be provided for each of these representations, based on a text that starts with the following sentences:

```
NIPPON SANSO K.K. HAS SET UP A JOINT VENTURE WITH A
TAIWANESE INVESTMENT FIRM IN MALAYSIA TO PRODUCE STAINLESS
THERMOS BOTTLES, THE LARGEST JAPANESE OXYGEN MANUFACTURER
SAID FRIDAY.

THE JOINT VENTURE, TOP THERMO MFG (MALAYSIA) SDN. BHD. IS
NIPPON SANSO'S SECOND PRODUCTION BASE IN ASIA FOLLOWING A
TAIWANESE PLANT, COMPANY OFFICIALS SAID.

THE MALAYSIAN COMPANY IS CAPITALIZED AT 19 MILLION RINGGIT,
OF WHICH NIPPON SANSO HAS PROVIDED 60 PCT AND TAIWAN'S KING
WARM INVESTMENTS LTD. 40 PCT.
```

4.1.1.1 All Phrases

One way of representing the problem is to present each new phrase along with *all* of the previous phrases in a text to the classifier; the classifier could then return the previous phrase – or set of phrases – which is [are] coreferent with the new phrase.

For example, consider the three sentences presented at the start of this section. If each instance represents a new phrase and all previous phrases, instances representing the following sets of phrases would be generated:

1. Current phrase: `NIPPON SANSO K.K.`
   Previous phrases: *none*
   Classification: *NIL*

2. Current phrase: `A JOINT VENTURE ...IN MALAYSIA`
   Previous phrases: `NIPPON SANSO K.K.`
   Classification: *NIL*

3. Current phrase: `A TAIWANESE INVESTMENT FIRM`
   Previous phrases: `NIPPON SANSO K.K.`
   `A JOINT VENTURE ...IN MALAYSIA`
   Classification: *NIL*

4. Current phrase: `THE LARGEST JAPANESE OXYGEN MANUFACTURER`
   Previous phrases: `NIPPON SANSO K.K.`
   `A JOINT VENTURE ...IN MALAYSIA`
   `A TAIWANESE INVESTMENT FIRM`
   Classification: *"NIPPON SANSO K.K."*

   and so on.

There are a number of difficulties with this approach. Important items in the news are often repeated in several different articles; however, most entities in a corpus of articles are distinct, i.e., they are not repeated [often] in other stories. Therefore, the set of classes, where each class represents a distinct reference to an entity, is potentially

quite large, and the prospects for any sort of general concept being captured by a machine learning algorithm are quite slim.

Another problem is that this approach would require a variable-length instance representation – the "size" of each instance would depend upon the number of previous phrases encountered in a text at any given point. Many of the best-known machine learning algorithms are set up to handle only fixed-sized instances.[2]

A third argument against this approach is based on cognitive plausibility: when people read a news article, it is unlikely that they consider *all* of the previous phrases as possible antecedents when they encounter each new phrase. While it is not necessary that an automated coreference resolution system work in a manner similar to human readers, a cognitively plausible model would be worth considering.

4.1.1.2 All Entities

It is unlikely that humans consider all previous references in a text when resolving a new reference; it is more likely that humans consider previous *referents* – internal representations of the *entities* previously referenced rather than the references themselves – although short-term memory probably limits this set to something less than *all* of the previous entities, particularly for long news articles.

Under this representation, a set of instances would represent the following sets of merged phrases (using the same example at the start of this section):

1. Current phrase: `NIPPON SANSO K.K.`
   Previous entities: *none*
   Classification: *NIL*

2. Current phrase: `A JOINT VENTURE ...IN MALAYSIA`
   Previous entities: `NIPPON SANSO K.K.`
   Classification: *NIL*

3. Current phrase: `A TAIWANESE INVESTMENT FIRM`
   Previous entities: `NIPPON SANSO K.K.`
   `A JOINT VENTURE ...IN MALAYSIA`
   Classification: *NIL*

4. Current phrase: `THE LARGEST JAPANESE OXYGEN MANUFACTURER`
   Previous entities: `NIPPON SANSO K.K.`
   `A JOINT VENTURE ...IN MALAYSIA`
   `A TAIWANESE INVESTMENT FIRM`
   Classification: *"NIPPON SANSO K.K."*

5. Current phrase: `THE JOINT VENTURE, TOP THERMO MFG (MALAYSIA) SDN. BHD.`
   Previous entities: `NIPPON SANSO K.K., THE LARGEST ...`
   `A JOINT VENTURE ...IN MALAYSIA`
   `A TAIWANESE INVESTMENT FIRM`
   Classification: *"A JOINT VENTURE ... IN MALAYSIA"*

   and so on.

[2] Exceptions include *inductive logic programming (ILP)* algorithms such as FOIL [Quinlan, 1990].

A representation that presents all previous entities along with a new reference to a classifier suffers from some of the same problems as the "all phrases" approach. Although the set of all entities referenced in a corpus is smaller than the set of all references, this is still a large number, and the prospects for general concepts emerging from a learning algorithm are still rather remote.

Similarly, while the set of all entities referenced in a text is smaller than the set of all references in a text, a representation that encodes all previous entities would still result in variable-length instances, although the instances may be smaller, and there may be less variability in the length of the instances.

4.1.1.3 One Entity

If each new reference were compared to one previous entity at a time, fixed length instances could be used to represent the problem – each instance would pair a reference with a previous entity and the classifier would return a binary classification: coreferent or not coreferent.

When instances are generated for `THE MALAYSIAN COMPANY` in the third sentence above, three entities will have been referenced:

1. The Japanese partner in the joint venture, which was referenced by two relevant phrases: `NIPPON SANSO K.K.` and `THE LARGEST JAPANESE OXYGEN MANUFACTURER`.

2. The Taiwanese partner in the venture, which was referenced by a single phrase: `A TAIWANESE INVESTMENT FIRM`.

3. The Malaysian company that was formed by the two partners, which was referenced by two relevant phrases: `A JOINT VENTURE ...IN MALAYSIA` and `THE JOINT VENTURE, TOP THERMO MFG (MALAYSIA) SDN. BHD.`.

If a system has correctly extracted the nationality of each of these entities, then the task of resolving the reference to `THE MALAYSIAN COMPANY` is straightforward, since there is only one company that is located in Malaysia.

This approach overcomes all of the problems of the previous approaches: it has a set of only two classes – coreferent and not coreferent – rather than a large (and potentially infinite) set of classes, and it would require fixed length instances rather than variable length instances.

One benefit of this approach is that information contributed by each of the previous references to the same entity would be *merged* together. If previous references provided information about the name and location of an entity, and a new reference contained a similar name or compatible location, then all of this information would be available to the classifier.

The problem with this approach is that it complicates evaluation. Instead of being able to construct all possible instances off-line, as would be the case for the "all phrases" approach, the instances must be constructed on-line as a text is processed. Depending on the complexity of the features that are extracted from each phrase (and from each set of merged phrases), this can be an expensive process.

A further complication arises from the dilemma of what to do about incorrect classifications. If references are merged as they are resolved, then one false positive classification could result in references to distinct entities being merged. This incorrect merging would result in errors in the construction of subsequent instances, potentially leading to a cascade of incorrect classifications.

4.1.1.4 One Phrase

The simplest approach to both representing the coreference resolution problem and simplifying subsequent evaluation is to pair each new reference with each previous reference in a text, yielding a new instance for each distinct pairing. Under this scheme, each instance would have a fixed length (two phrases) and either a positive (coreferent) or negative (not coreferent) classification.

1. Current phrase: `A JOINT VENTURE ...IN MALAYSIA`
   Previous phrase: `NIPPON SANSO K.K.`
   Classification: *NO* (not coreferent)

2. Current phrase: `A TAIWANESE INVESTMENT FIRM`
   Previous phrase: `NIPPON SANSO K.K.`
   Classification: *NO*

3. Current phrase: `A TAIWANESE INVESTMENT FIRM`
   Previous phrase: `A JOINT VENTURE ...IN MALAYSIA`
   Classification: *NO*

4. Current phrase: `THE LARGEST JAPANESE OXYGEN MANUFACTURER`
   Previous phrase: `NIPPON SANSO K.K.`
   Classification: *YES* (coreferent)

5. Current phrase: `THE LARGEST JAPANESE OXYGEN MANUFACTURER`
   Previous phrase: `A JOINT VENTURE ...IN MALAYSIA`
   Classification: *NO*

6. Current phrase: `THE LARGEST JAPANESE OXYGEN MANUFACTURER`
   Previous phrase: `A TAIWANESE INVESTMENT FIRM`
   Classification: *NO*

   and so on.

Evaluation would be simplified in two ways. All instances could be constructed off-line, greatly decreasing the time required for each evaluation. This can be especially important during system development, since a quick response (on the order of minutes) can provide feedback that can be used to refine existing features or add new ones – evaluations that require hours (or days) to run are burdensome to this process.

Furthermore, the problem of the cumulative effects of incorrect merging is eliminated – an early misclassification need not result in a cascade of future misclassifications.

Of course, without merging references as classification proceeds, all the information about a particular entity is scattered among the different previous references to that entity. However, the information contained in one (or even several) reference(s) is often sufficient for establishing a coreference link between a new reference and a previous reference.

For example, the best match for `NIPPON SANSO` in the third sentence above is going to be `NIPPON SANSO K.K.`; the classifier may not be able to link the phrase with `THE LARGEST JAPANESE OXYGEN MANUFACTURER`, but if it had earlier linked the first two references to Nippon Sanso, then transitive closure can be used to merge the three phrases.[3]

[3] See Section 6.3 for an elaboration on the issue of transitive closure in the coreference relation.

Table 4.1 Sample Feature Vector

Attribute	*Value*
PRONOUN-1	*NO*
PRONOUN-2	*NO*
JV-CHILD-1	*NO*
JV-CHILD-2	*UNKNOWN*
SAME-SENTENCE	*YES*
ALIAS	*NO*
COMMON-NOUN	*NO*
COMMON-LOC	*NO*

4.1.2 Feature Vectors of Attribute/Value Pairs

Once a representation for the problem has been chosen, a scheme for encoding instances of the problem must be selected. A common format for presenting instances to a machine learning algorithm is a *feature vector* wherein each vector position represents an *attribute name* and each vector element represents the *value* corresponding to that attribute.

As an example, consider the last pair of phrases in the previous section: `A TAIWANESE INVESTMENT FIRM` and `THE LARGEST JAPANESE OXYGEN MANUFACTURER`. A feature vector representing a small subset of the features described in the experiments reported in Chapter 8 is shown in Table 4.1.[4]

4.2 The C4.5 Machine Learning Algorithm

The C4.5 machine learning algorithm was selected for the experiments reported in this dissertation because the algorithm is clearly explained [Quinlan, 1993], an implementation is readily available[5] and the system is widely used.

RESOLVE is written in Common LISP, as is the information extraction system for which it was originally intended to be used as the coreference resolution component. C4.5 is written in C. In order to eliminate the need for cross-platform infrastructure[6], the C4.5 tree induction, pruning and classification procedures were re-implemented in Common LISP for the work described in this dissertation.

[4]The meaning of the attribute names is described in Section 7.2.2. As a brief explanation, the feature vector represents the following facts: the first phrase is not a pronoun; the second phrase is not a pronoun; the first phrase does not refer to a joint venture company (it refers to a ***parent*** of a joint venture); the second phrase may or may not refer to a joint venture company; the two phrases come from the same sentence; the second phrase is not an alias of the first phrase; the two phrases do not share a common noun; and the two phrases do not share a common location.

[5]A diskette containing the complete implementation is available from Morgan Kaufmann Publishers for a nominal fee.

[6]For example, foreign function calls from the Common LISP environment.

4.2.1 Decision Tree Induction

C4.5 is an algorithm that creates a decision tree classifier from a collection of instances, each represented as a vector of attribute-value pairs and a class label. A complete description of the tree construction algorithm can be found in Quinlan [1993], Chapter 2; the following provides a simplified account, assuming two classes (positive and negative) and discrete valued attributes.

For a set of training instances T and a pair of classes $C_1, C_2,$, a decision tree is constructed as follows:

1. If all the instances in T are labeled with the same class, C_k, a decision tree consisting of a single (*leaf*) node is constructed, identifying the class as C_k.[7]

2. Otherwise, an attribute A_i is selected which has possible values $V_{i,1}, V_{i,2}, \ldots, V_{i,n}$ and the instances are partitioned into subsets $T_1, T_2, \ldots, T_n$ according to the value of that attribute in each instance; a decision tree is constructed with A_i as the root node, and a branch for each of the possible values ($V_{i,j}$) of A_i; the process is recursively applied to the subset of instances (T_i) associated with each branch in the new tree.[8]

4.2.2 Decision Tree Pruning

Decision trees often *overfit* the data with which they are trained, i.e., they tend to make spurious distinctions based on the training sample that are not likely to hold for the larger population from which the sample is drawn.[9] In order to compensate for this tendency, C4.5 can employ a *pruning* algorithm to simplify its initial decision trees.

The C4.5 pruning algorithm works in a bottom-up fashion: it descends the decision tree and on its way back up, at each node N, it estimates which situation would be most likely to result in the fewest errors when classifying unseen instances:[10]

- Replace the subtree at N with a leaf node labeled with the most frequent class of the instances represented by N,

- Replace the subtree at N with its largest child, i.e., the child of N that represents the largest number of instances, or

- Retain N in its current form

[7] If there are no instances, i.e., $|T| = 0$, the class label of the parent node is returned. There is also a threshold such that if all but d instances are in the same class C_k, then a leaf node labeled with C_k is returned.

[8] The attribute is selected so as to maximize the mutual information between the test of that attribute and the class distribution of the instances in each of the partitions. The mutual information measure – called the ***gain criterion*** – can normalized to compensate for the fact that attributes with many possible values tend to have disproportionately higher ***information gain*** than attributes with few possible values; the application of this normalization to the gain criterion is called the ***gain ratio criterion***.

[9] Actually, this tendency to overfit training data is endemic to most machine learning algorithms, which nearly always are given a sample which may or may not be representative of the larger population.

[10] The pessimistic estimate of errors at a decision tree node lies between a pair of confidence limits based on the binomial distribution. Quinlan [1993], Chapter 4, provides a more complete explanation of the decision tree pruning procedure employed by C4.5.

IF	$A_1 = V_{1,i}$
AND	$A_2 = V_{2,j}$
...	...
AND	$A_m = V_{m,k}$
THEN	$class = X$

Figure 4.1 Format of C4.5 Production Rules

Whichever action would result in the lowest error estimate at N is taken, and the pruning procedure is repeated as the decision tree is traversed.

A pruned decision tree is often far smaller, more comprehensible, and more accurate on unseen instances than its unpruned counterpart. Therefore, the decision trees used by RESOLVE throughout the experiments reported in this dissertation were all pruned using the normal C4.5 pruning procedure.[11]

4.2.3 Decision Tree Classification

Once a decision tree has been constructed from a set of training instances, it can be used to classify new, unseen instances in the following way, where the *current node* is initialized to the *root node* of the decision tree:

1. If the current node of the decision tree is a *leaf*, return the class label at that node as the classification for the instance

2. Otherwise, let A_i be the attribute tested at the current node of the decision tree, and let $V_{i,j}$ be the value of that attribute in the instance being classified.[12] Find the branch associated with value $V_{i,j}$ and make the node at the end of that branch the new current node, and repeat the procedure.

4.2.4 Production Rules

C4.5 also includes a mechanism for generating a set of production rules directly from a decision tree. These production rules have the general form shown in Figure 4.1.

The A_i and $V_{i,j}$ represent the same set of attributes and values that are used to construct the decision tree from training instances. Each clause (attribute/value pair) in the antecedent (or *if* portion) of a rule represents the same information that can be found in a single branch of a decision tree. The C4.5 rule induction procedure is described in Quinlan [1993], Chapter 5.

Initial experiments with the C4.5 rule induction system on instances created for the coreference resolution task produced long lists of rules, many of which did not correlate well with intuitions about what such rules should look like. Since the pruned

[11] There are a number of parameters that can be adjusted to affect the behavior of the pruning algorithm; the default values for these parameters were used in each case. Section 10.3.3 will discuss some issues involved in pruning and propose some future work on procedures that are more sensitive to the type of evaluation typically done for coreference resolution (coreference evaluation will be discussed in Chapter 6).

[12] The value of attribute A_i may be unknown for the current instance; the treatment of unknown values is discussed in greater detail in Section 7.2.4.1.

decision trees were both more compact and more comprehensible, the decision tree representation is used throughout all the experiments reported in this dissertation.

4.3 Machine Learning and Natural Language Processing

RESOLVE joins a growing list of systems that incorporate machine learning techniques in attempting to solve problems in NLP. One of the major motivations behind this trend is the widespread availability and use of large corpora for evaluating NLP systems, which has allowed more researchers to focus their efforts on *corpus-based* NLP. Early NLP systems were often tested on a small handful of sentences that were constructed specifically for the purpose of testing some aspect of language understanding; more recent systems are tested on a large set of sentences and texts that are drawn from a corpus of texts that were written for other purposes, e.g., newspaper articles written for human readers.

The availability of corpora allows NLP researchers to focus on naturally occurring language. The corpora also provide a rich source of training material, an important requirement for the application of machine learning techniques. Some corpora include extensive annotations and can be used directly for training, as was the case with the part-of-speech and bracketing annotations of Wall Street Journal articles from the Penn Treebank [Marcus *et al.*, 1993] being used to induce statistical parsers such as those developed by Magerman [1994, 1995] and Collins [1996], or the named entity and coreference link annotations of another, much smaller set of Wall Street Journal articles that formed part of the corpus used for the MUC-6 evaluation.

Other corpora include less structure and typically need to be modified for use as training material, as was the case with the MUC-4 corpus of newswire stories and their associated key templates. In order to use this material for training, some mechanism must be developed for linking information contained in the key templates to their source texts, as was done for the AUTOSLOG dictionary construction tool [Lehnert *et al.*, 1992, Riloff, 1993].

Machine learning techniques have been applied to problems that span the spectrum of research in natural language processing. Some systems use machine learning to solve problems at the level of *sentence analysis*, i.e., the analysis of individual sentences in isolation. Problems at this level include part-of-speech tagging, semantic feature tagging, prepositional phrase attachment and syntactic analysis of the entire sentence.

More recently, some work has been done on using machine learning for solving problems in *discourse analysis*, i.e., the analysis of phenomena that occur in a sequence of sentences (or utterances). Machine learning techniques have been applied to problems such as the identification of discourse segment boundaries and coreference resolution.

The following sections will describe some of the previous research that has been done in applying machine learning to problems in sentence analysis (Section 4.3.1) and problems in discourse analysis (Section 4.3.2).

4.3.1 Machine Learning for Sentence Analysis

There are several systems that are *trained* to do part-of-speech tagging, i.e., assigning part-of-speech labels (e.g., noun, preposition, past-participle verb) to each word in a sentence. Some of these systems [Church, 1988, Weischedel *et al.*, 1993] use statistical methods in conjunction with a corpus of sentences in which each word has

been assigned a unique part-of-speech tag; such systems typically determine the probability that a part-of-speech tag (T_i) should be assigned to a particular word (W_i) in a sentence, based on either a *bigram* model of the prior probabilities of that tag and the previous tag ($P(W_i|T_iT_{i-1})$) or a *trigram* model of the prior probabilities of that tag and the previous two tags ($P(W_i|T_iT_{i-1}T_{i-2})$).

Brill developed a technique called *transformation-based learning* and applied that technique to the problems of part-of-speech (POS) tagging [Brill, 1994] and prepositional phrase (PP) attachment [Brill and Resnik, 1994]. In Brill's work, an initial solution to a problem (labeling a word with a POS tag or attaching a PP) is proposed using a very simple method, and then a transformation is applied in contexts where the method has produced errors in the past. Brill's POS tagger initially labels each word in a sentence with its most frequent POS tag, which results in a correct labeling 90% of the time. It then applies a set of *transformation rules* that specify a context in which to change an old tag to a new tag. The context that can be examined by these rules consists of a window of 5 words and tags: when considering a transformation of a tag t_i initially assigned to word w_i, the rule antecedents include tests that examine some combination of the preceding and succeeding tags and words ($t_{i-2}, \ldots, t_{i+2}$ and $w_{i-2}, \ldots, w_{i+2}$).[13]

The transformation rules are learned from examining the tagging errors – where a word was assigned a tag t_i but should have been assigned the tag t_j – made by the "most frequent tag" algorithm (plus any previously learned rules) on a corpus of POS-tagged sentences.[14] A set of possible rules is proposed to correct the t_i/t_j errors, these rules are evaluated by applying each one to the tagged sentences, and the rule that results in the largest reduction in errors is added to the current set of transformation rules. The sentences in the corpus are tagged again, and the set of transformation rules is used to change some of those tags. This process is repeated until no proposed transformation rule reduces the error. An example of a transformation rule that is learned by the system is to change a *preposition* POS tag to an *adverb* POS tag whenever the word two positions to the right is "`as`" – this rule corrected for sequences such as "`as tall as`," which is assigned the tag sequence *adverb adjective preposition* in the corpus.

Cardie [1993] compared three different machine learning methods – case-based learning (CBL), decision trees and a hybrid approach combining CBL and decision trees – to learn the part-of-speech label and semantic classes (general and specific) for an unknown word, based on the context surrounding the unknown word. An instance contained 20 attributes representing local information (a word, part-of-speech label, general semantic class and specific semantic class) from a window of 5 words in a sentence – the current word, the two previous words and the two following words – and 13 attributes representing global information from the current sentence – information about the subject, verb and direct object of that sentence. Cardie found that a hybrid approach – using a decision tree to learn the best features to use for retrieving previous cases, then using a *K nearest neighbor (k-NN)* algorithm to select a case which is used to fill in the part-of-speech or semantic feature labels for a new word – outperforms systems that use only decision trees or only CBL.

The important thing to note about applications of machine learning to problems that occur at the sentence analysis level is that the scope of such problems is limited to a single sentence, unlike problems in discourse analysis for which there are no easily defined boundaries. The features defined for sentence-level problems can make use of this limited scope; for discourse-level problems, additional features are often required which extend beyond the scope of isolated sentences.

[13] For unknown words, additional factors such as suffixes were available to the rules.

[14] The Penn Treebank corpus of Wall Street Journal articles [Marcus *et al.*, 1993].

4.3.2 Machine Learning for Discourse Analysis

Machine learning techniques have also been used for discourse analysis, i.e., problems in natural language processing that extend beyond individual sentence boundaries to inter-sentence phenomena. One system, described below, used machine learning to handle *all* of the discourse analysis required by an information extraction system. Two other areas of discourse analysis that have benefited from the use of machine learning techniques are discourse segmentation and coreference resolution; examples of work in these areas will also be described in this section.

4.3.2.1 Discourse Analysis for Information Extraction

Soderland and Lehnert [Soderland and Lehnert, 1994] used a large set of ID3 decision trees [Quinlan, 1986][15] to learn how to perform essentially all of the discourse processing functions required by an information extraction system. WRAP-UP constructed a separate decision tree for each type of entity and each type of relation among entities that was defined as relevant for the MUC-5 English Microelectronics task [Lehnert *et al.*, 1993]. One of the five subtasks WRAP-UP learned was when to merge different descriptions of the same entity, i.e., the coreference resolution task.

WRAP-UP differs from RESOLVE in a number of ways. WRAP-UP was applied to a domain that was very different from both the MUC-5 English Joint Ventures domain and the MUC-6 corporate management changes domain, the two domains to which RESOLVE was applied; coreference resolution is less important to successful information extraction in the EME domain than it is in the two domains to which RESOLVE was applied.[16] WRAP-UP instances are generated from the output of the CIRCUS sentence analyzer[17] and the key templates that represent the information to be extracted from a given text; errors from the sentence analyzer and key templates affected the performance of WRAP-UP on all five subtasks. The output of WRAP-UP was a response template; the system was evaluated in terms of how well it performed on the overall information extraction task, but no quantitative assessment was made of its performance on the five individual subtasks, such as coreference resolution.[18]

4.3.2.2 Discourse Segmentation

Litman and Passonneau [1996] used C4.5 to learn how to identify discourse segment boundaries, i.e., boundaries between adjacent phrases that reflect different speaker intentions in a narrative. Instances were composed of 12 attributes extracted from each phrase in a narrative; these features fell into four general categories: prosody (pauses and punctuation), cue phrases (words indicating a new speaker intention), noun phrases (explicit or implicit coreferential relationships among noun phrases in different prosodic phrases) and one meta-feature that combined elements from the prosodic and cue phrase features. The values for some of the attributes were generated automatically from syntactic and lexical analysis of the phrases; the values

[15] ID3 was the predecessor of C4.5.

[16] Stephen Soderland, personal communication.

[17] As it was configured for the MUC-5 EME task [Lehnert *et al.*, 1993].

[18] In fact, given the complexity of the overall information extraction task, quantitative assessment for the individual discourse subtasks, such as coreference resolution, is all but impossible. This difficulty in assessing subtasks was one of the motivations behind the development of four separate subtasks in the MUC-6 evaluation, which is discussed in Section 2.3.2 and in greater detail in Appendix C.

of other attributes (including one that denoted coreferent relationships between noun phrases in different phrases) were based on manual annotations of the narratives.[19] Each instance was labeled with one of two class labels, *boundary* and *non-boundary*, indicating whether the phrase in question was the first phrase in a new discourse segment (i.e., reflected a different speaker intention than the preceding phrase).

Two aspects of Litman and Passonneau's work are of particular relevance to the current work. One is that the motivation behind applying machine learning to the problem of discourse segmentation was the daunting complexity of manually determining the best possible combination of features.[20] The other is that they discovered that the C4.5 decision trees achieved better performance than their hand-crafted algorithms.

4.3.2.3 Coreference Resolution

Connolly, Burger and Day [1994] conducted an experiment comparing the performance of a hand-crafted algorithm for coreference resolution that was composed of a decision list of 50 rules to a variety of machine learning algorithms.[21] They showed that both a C4.5 decision tree and neural network could outperform their hand-crafted algorithm, a result similar to that reported in Chapter 7. However, they used a single measurement of classification accuracy to evaluate their systems; Chapter 6 argues that *recall* and *precision* provide more information about the performance of a coreference resolution system. The set of attributes used in their experiments was quite small, testing only 7 features of the phrases that comprise their instances;[22] the set of attributes used in the experiments reported in Chapters 8 and 9 is much more extensive.

The experiments reported in Connolly, *et al.*, differ from the experiments in this work in a number of other ways:

- Source of Training and Testing Data
  - The instances used in training and testing RESOLVE were generated from a set of annotations of texts via the CMI interface; these annotations included information that would have normally come from a sentence analyzer. This system-mediated annotation method was used in order to simplify the credit-assignment problem – any errors made by RESOLVE can be ascribed to the feature set, training or learning algorithm.
  - The instances used in training and testing the systems reported in Connolly, *et al.*, were based on annotations of the output of a sentence analyzer rather than annotations on the texts; the data therefore included whatever

[19] The annotation methodology is described in Passonneau [1994].

[20] This complexity is based on a set of 12 features, a number which may not seem overwhelming for other tasks to which machine learning has been applied. Note that the determination of discourse segment boundaries remains a difficult problem for which no adequate, comprehensive theory has yet been developed, much less implemented.

[21] The machine learning algorithms tested include a posterior classifier, a simple Bayes classifier, a C4.5 decision tree and a neural network, plus a number of hybrid systems that combine different elements of these individual approaches.

[22] The authors note the small number of attributes and express their intention to perform later experiments involving a larger set of attributes.

errors were generated by the sentence analyzer.[23] These annotations were done manually, possibly introducing another source of errors.

- Instance Representation

  - An instance for RESOLVE is a set of features extracted from a *pair* of phrases – the first element of the pair is a "new" phrase encountered in a text, and the second element of the pair is a phrase that precedes the new phrase in the text – and a class label indicating whether the two phrases are coreferent or not. The set of phrases in a text for which instances are created is constrained by their relevancy to a predefined information extraction task. However, instances are created for all possible pairings of relevant phrases, so that RESOLVE is expected to determine whether a new phrase is either coreferent with an earlier phrase or is the first reference to a new entity.
  - An instance for the systems tested in Connolly, *et al.*, can be viewed as a set of features extracted from a 3-tuple or *triple* of phrases – a "new" phrase and two phrases that precede the new phrase in the text – and a class label indicating which of the two preceding phrases is the most likely antecedent for the new phrase. No relevancy constraint was mentioned in the paper, so all phrases may have been considered. However, instances are created only for new-phrases that are, in fact, coreferent with some preceding phrase, i.e., the classifiers used in Connolly, *et al.*, are expected to find the correct antecedent and not to determine whether a given phrase has any antecedent.

- Classification Procedure

  - For each new phrase encountered in a text, RESOLVE is presented a sequence of instances representing pairings of the new phrase with each preceding phrase in that text; each pair is presented to RESOLVE until a positive classification is returned (or there are no more pairs to consider, in which case the new phrase is interpreted to be the first reference to an entity).
  - In the systems tested in Connolly, *et al.*, a classifier is presented a triple <new-phrase, old-phrase-1, old-phrase-2>; one of the old phrases, old-phrase-*i*, is selected as the most likely antecedent of the new phrase, and then a new triple is constructed with the new phrase, old-phrase-*i* and another preceding phrase. This process is repeated for all of the preceding phrases. The most likely antecedent selected from the last triple "wins", i.e., it is interpreted as the antecedent for the new phrase.

Aone and Bennett [1995] have also used C4.5 for coreference resolution, and have shown that their Machine-Learning Resolver (MLR) machine learning system can outperform their Manually-Designed Resolver (MDR) in the domain of Japanese Joint Ventures (JJV). As was the case for RESOLVE, the instances created for the MLR represent pairs of phrases, where one element of the pair is an anaphor (or "new phrase") and the other element is a possible antecedent (or "preceding phrase");

[23] No mention is made as to whether any system output was discarded due to serious errors; the potential problems involved in attempting to use sentence analyzer output are discussed at greater length in Section 5.4.1.

unlike RESOLVE, the instances represented only those new phrases that had already been marked as coreferent with some preceding phrase.

The performance of the MDR was compared to the performance of the MLR on the same data set according to two different evaluation metrics;[24] three different parameters of the MLR were varied for a total of six different parameter settings. The MLR achieved higher performance according to one evaluation metric, but slightly lower performance according to another, for all parameter settings; however, for certain classes of anaphoric reference, roughly corresponding to aliases and definite references, the MLR achieved higher performance according to both metrics. This work shows that a machine learning system can achieve good performance on coreference resolution for Japanese, however it is not clear whether similar results would hold in English[25]. Since only seven of the 66 features used are listed in the paper, and no definitions are provided, it is not clear how many of the features used in this experiment are specific to the Japanese language, the joint ventures domain, or the SOLOMON sentence analyzer. RESOLVE is currently language-dependent (English), its domain-specific features are carefully delineated from its domain-independent features (see Chapter 8), and it does not rely on the output of any particular sentence analyzer.

[24] These two metrics – *recall* and *precision* – will be described in detail in the next chapter.

[25] Although Chapter 7 shows that a machine learning system can achieve performance comparable to a manually engineered rule-based system for coreference resolution on texts from the English Joint Ventures domain.

CHAPTER 5

COLLECTING THE DATA

The development of any coreference resolution system requires a set of examples of phrases that are coreferent, and preferably examples of phrases that are not coreferent. The examples provide a source of ideas about which features of the phrases are important for coreference resolution, whether these features are used in a manually engineered system or a trainable system. For a trainable system, such examples are also necessary for the training and testing of the concept induced by the learning algorithm.

What is the source of the example phrases? Which phrases are candidates for coreference resolution? What constitutes an example of coreferent phrases? What constitutes an example of non-coreferent phrases? Where should examples of coreferent and non-coreferent phrases come from? How should the examples be collected – manually, automatically or something in between?

This chapter will provide one set of answers to these questions by describing the approach to collecting examples taken for the development of RESOLVE.

5.1 Source of Phrases

Much of the earlier research on coreference resolution was based on sentences created or carefully selected by the researchers looking into particular aspects of the problem. For example, Hobbs [Hobbs, 1978] illustrated many of the aspects of his pronoun resolution algorithm using sentences he created for that purpose,[1] and Brennan, Friedman and Pollack [1987] (hereafter referred to as BFP), created two short stories, each containing four simple sentences, to demonstrate their centering approach to pronoun resolution.

Specially created or selected sentences are useful for illustrative purposes – good examples often render complex algorithms more understandable. This approach suffers from some drawbacks, though. One problem is that constructs that are used to demonstrate aspects of particular algorithms may or may not be representative of the types of constructs that frequently occur in naturally occurring texts. Algorithms that successfully process interesting, but infrequent, constructs may not be very useful in realistic language processing tasks such as information extraction. The simple constructs used for illustration in BFP, for example, are rarely found in any relevant sentences from MUC-5 EJV texts.

Another problem with the use of sentences that are constructed for the sole purpose of testing an algorithm is that the use of such sentences hinders a comparative evaluation of different algorithms (since everyone is constructing different sentences for different algorithms). Walker [1989] focused on three different sources of pronominal references in her comparison of the Hobbs algorithm and the centering (or BFP)

[1] Hobbs also includes sentences created as examples for illustrating previously developed algorithms.

algorithm, enabling her to draw some conclusions about the relative strengths and weaknesses of each approach.

Research in many areas of NLP is moving in the direction of corpus-based approaches. A large collection of texts is more likely to be more representative of a broad range of linguistic phenomena than a small set of sentences. Evaluation of different algorithms is easier when the algorithms have been developed and tested on the same corpus.

The focus of the current work is on the MUC-5 EJV corpus, a collection of 1000 news articles that mention business tie-ups among two or more organizations.[2] The articles were drawn from 74 different news organizations[3], and cover the eleven year period from 1980 through 1991[4]. This corpus is constrained with respect to its genre (news articles) and its domain (business joint ventures), but the texts contain a vast array of linguistic constructs.

The constraints provided by the MUC-5 EJV task definition help make the knowledge requirements for any discourse-level language processing task more tractable. However, even within the constraints dictated by the choice of corpus, the coreference resolution task remains quite challenging.

5.2 The Focus on Relevant Phrases

The information extraction orientation of the coreference resolution work undertaken in this dissertation helps to constrain the problem in two important ways. Information extraction systems extract specific types of information about specific entities, and specific relationships among these entities; this means that many phrases, which contain information that is not relevant to the particular information extraction task, can be completely ignored. Furthermore, some phrases that *do* refer to relevant entities do not contribute any information specified by the task, and so these phrases can likewise be ignored.

5.2.1 Relevant Entities

News articles about business joint ventures often contain many facts relating to parent companies involved in the ventures as well as the new company formed as part of the venture. The MUC-5 task definition listed a set of entities and relationships that information extraction systems were expected to extract from the texts. The relevant types of entities for this task included organizations involved in a joint venture, people associated with these organizations, facilities used by the venture, and the products or services provided by the venture. The relationships among the organizations (e.g., parent, child, partner) and the proportions owned by each parent organization are among the relevant relationships specified for this domain.

5.2.2 Relevant References

Entities that are relevant to the MUC-5 task are often referenced in several places throughout a text. Some of these references contribute new, relevant information

[2] The organizations involved in a tie-up were usually companies, but were sometimes governments and occasionally individual people.

[3] Approximately half of the texts were drawn from four news organizations.

[4] Approximately half of the texts cover the three year period from 1989 through 1991.

specified by the task, e.g., the name of an organization or the location of a facility. Other references, though, contribute information not relevant to the task, e.g., the closing price of a parent company on the New York Stock Exchange, which is mentioned in many of the Wall Street Journal articles.

Since irrelevant references were not likely to be identified by an information extraction system and then presented to RESOLVE for coreference classification, such references were not included as instances used by RESOLVE during its training or testing for the experiments reported in Chapters 7 and 8.

5.3 Other Constraints on Phrases

In addition to constraints based on notions of relevancy that are defined for an information extraction task, other constraints were used to make the coreference resolution task more manageable.

5.3.1 Noun Phrases vs. Modifiers

There were some noun phrases that had sub-phrases (or individual words) which referred to some other organization. For example, in `FORD MOTOR CO.'S EUROPEAN UNIT`, the sub-phrase `FORD MOTOR CO.` refers to something other than the company's `EUROPEAN UNIT`. Another type of situation in which this sometimes occurs can be seen in the phrase `TAMOTSU GOTO, JAL SENIOR VICE PRESIDENT`, where the sub-phrase `JAL` refers to the company of which Tomatsu Goto is a senior vice president.

These sub-phrases were not extracted by the interface since, at the time the data was collected for the EJV domain, the sentence analyzer with which it was to be used had no capability for extracting such sub-phrases.[5] As was mentioned earlier, the intent of the interface was to extract only the types of information for which it was reasonable to expect that an automated system could.

5.3.2 Singular Noun Phrases

Only singular noun phrases, i.e., noun phrases that refer to single entities rather than to sets of entities, are considered candidates for coreference resolution. This constraint was imposed due to the additional complexity of processing *multi-referent* phrases (noun phrases that refer to sets of entities).

Some multi-referent noun phrases refer to conjunctions of singular noun phrases, as in the following two sentences:

```
THREE JAPANESE FIRMS HAVE SIGNED AN AGREEMENT WITH A MAJOR
CZECH BANK TO SET UP A JOINT VENTURE TO BUILD HOTELS AND
SHOPPING CENTERS IN CZECHOSLOVAKIA, THE JAPANESE PARTNERS
ANNOUNCED TUESDAY.

THE PARTIES INVOLVED ARE THE SLOVAK STATE SAVINGS BANK (SSB)
FROM THE CZECH SIDE, AND TRANS-MEDIA RESOURCES INC., NEXAS
AND SAKATA PURIFIED CO. FROM JAPAN, AS WELL AS
CZECH-AMERICAN ENTREPRENEUR ANTON KAJRICH.
```

In this example, the phrases `THREE JAPANESE FIRMS` and `THE JAPANESE PARTNERS` both refer to three of the five organizations involved in the joint venture:

[5] Such extraction was done near the end of processing by special purpose heuristics.

- TRANS-MEDIA RESOURCES INC.
- NEXAS
- SAKATA PURIFIED CO.

Another phrase, THE PARTIES, refers to all five entities involved in the joint venture. Determining which phrases refer to multiple entities, determining the number of entities to which such phrases refer, and then determining the antecedents of such phrases are all difficult problems.

The phrases THREE JAPANESE FIRMS and THE PARTIES are *multi-referent phrases*, i.e., they refer to multiple entities; due to the additional complexity of dealing with such phrases, they are excluded from the experiments reported in this dissertation.[6]

5.4 Methods for Collecting Phrases

The fastest method for collecting phrases relevant to an information extraction task is to use a sentence analyzer to identify such phrases automatically; however, this approach is viable only if a good sentence analyzer is available (and tuned to the domain), and even the best sentence analyzers usually generate errors in their output. The slowest approach to phrase collection is to identify phrases of interest manually; unfortunately, while human judgment is superior to automated methods, transcription errors are bound to occur, and it could take quite a long time to accumulate a large set of examples by hand.

A method that combines aspects of automatic and manual approaches was used for collecting phrases for the research done in this thesis. An interface was constructed to enable a human annotator to use his or her judgment in selecting relevant phrases, and to specify information that the human could easily infer about these phrases.

Each of these methods will be described in more detail below.

5.4.1 Automatic Methods

The NLP group at the University of Massachusetts first attempted to apply decision trees to the coreference resolution problem during the closing weeks of the MUC-5 development effort. For each text, the relevant phrases extracted by the CIRCUS sentence analyzer were paired, and each pair was presented to a user for classification. Due to errors in upstream processing, one of a set of three possible classifications was permitted for each pair:

- Coreferent: The pair of phrases was used to form a positive instance of coreference.
- Non-coreferent: The pair of phrases was used to form a negative instance of coreference.
- Discard: One, or both, of the phrases was "noisy" – irrelevant, improperly delimited (too short or too long), or mistagged with incorrect semantic features or other conceptual information.

[6]Other researchers working on coreference resolution have excluded such phrases for similar reasons, e.g., Suri [1993].

5.4.2 Manual Methods

One alternative to relying on system-generated output would be to annotate all the relevant phrases manually, perhaps with the aid of a text editor. The primary advantage to this method of collecting data is that it eliminates any errors that might be generated by a sentence analyzer.

The two main disadvantages of manually collecting examples are that

1. The annotator must make certain assumptions about which phrases would likely be identified by an automated system, as well as what slot-fill information could reasonably be inferred about any given phrase, and

2. *Unaided* manual collection be tedious and error-prone – in fact, it may introduces a new source of errors.

The first problem is unavoidable in any attempt to build a system that is independent of any sentence analyzer: if one does not rely on system output to extract examples, then one has to rely on human judgment as to which phrases might be extracted by a sentence analyzer and what additional information is likely to be identified during sentence analysis. The second problem can be greatly alleviated through the use of an intelligent interface for extracting examples; one that constrains the types of information collected about phrases.

5.4.3 A System-mediated Method: CMI

A graphical user interface, CMI (Coreference Marking Interface), was created for the data collection effort undertaken for this thesis. CMI enables an annotator to view a text, identify phrases in that text, specify the type of the entity to which the phrase refers (e.g., *organization* or *facility*), and fill various slots with values (e.g., *name* or *location*). The output from the interface is a *slot-and-filler* representation[7] of all the relevant references in a text.

The purpose of CMI is to enable a user to annotate a set of phrases in a text for the purpose of coreference resolution. The top level of the interface permits the user to develop and maintain a collection of sets of coreferent phrases, including the capabilities to:

- *Add* new phrases.
- *Copy* existing phrases (multireferent phrases will often appear in more than one sublist, i.e., in more than one set of coreferent phrases).
- *Edit* existing phrases, e.g., modify the slot information associated with a particular phrase.
- *Delete* existing phrases.

Some of the important details of CMI will be described in the following sections.

[7] A slot-and-filler representation consists of a set of labeled fields (***slots***) and values (***fillers***) for those fields. See Bobrow and Winograd [1977] for a much more comprehensive description of a slot-and-filler representation.

5.4.3.1 Entities

Any system for coreference annotation must both identify phrases and represent the coreference links among such phrases. One possible method for organizing such annotations is to associate a set of pointers with each reference, with each pointer uniquely identifying another reference with which the current reference co-refers.

The approach taken with CMI is to organize references around the entities to which they refer, i.e., group together all coreferent references. This approach offers two advantages over the previous approach:

1. Most relevant entities are referred to more than once – for example, 65% of *organizations* mentioned in the EJV texts are referenced more than once, with an average of 2.3 references to each *organization*. Structuring the representation around entities rather than individual references imposes an extra level of abstraction, reducing the number of objects that must be considered by the annotator at each level.

2. The groups of references form equivalence classes. Without this organization, such equivalence classes must be [re]computed each time they are needed. The equivalence classes represent the transitive closure of coreferent phrases; the importance of this transitive closure will be discussed in Section 6.3.

5.4.3.2 References

Phrases are selected by dragging the mouse cursor across a region of text. A parameter specifies whether the selected region is automatically expanded to word boundaries, which can simplify the selection process by permitting the annotator to be less exact in his or her mouse dragging.

Once the annotations for a phrase are completed, the phrase is highlighted in bold, and if the phrase is specified as relevant, it is also underlined.[8] The user can scroll through previously annotated phrases by selecting the buttons labeled *Next* or *Previous*, causing the next or previous phrase to be highlighted; the window will be adjusted if necessary so that the entire phrase is visible. The user can also directly select a previously annotated phrase by moving the cursor to a position within the phrase.

5.4.3.3 Syntactic Information

Since the data is collected without the aid of a sentence analyzer, the user is permitted to specify the syntactic information normally generated by this system component. Some of the syntactic information that may help with coreference resolution includes:

- *Discourse Segment*

  Nearly all of the short news articles in the EJV domain focus on a single main topic, e.g., providing information about a single joint venture. However, even some of these short articles may include multiple discourse segments, representing different secondary topics, e.g., one paragraph may contribute information primarily about the joint venture while another may contribute information only about the parent companies.

[8] Non-relevant phrases to relevant entities were also marked, to enable measurement of the effect of including non-relevant phrases among the candidates for coreference resolution.

Discourse segment information may be useful for coreference resolution [Passonneau, 1996]. However, identifying discourse segment boundaries is a notoriously difficult problem [Passonneau and Litman, 1993]; even humans often disagree about what constitutes a discourse segment or where a boundary exists between two segments. Therefore, discourse segment information was not included in the annotations used for current work.

- *Paragraph Index*

  Paragraph boundaries are easier to identify than discourse segment boundaries. The knowledge of whether two phrases come from the same paragraph or different paragraphs may be useful for coreference resolution, and may be an approximation to the sort of information that might be provided by annotating discourse segments. The paragraphs in a text could be numbered and then an integer (or index) indicating the paragraph from which a phrase was extracted could be associated each annotated phrase. However, knowledge about the paragraphs from which the phrases were extracted did not appear particularly useful in reading EJV texts, and so a feature to identify paragraph boundaries was not included in the interface.

- *Sentence Index*

  Two phrases from the same sentence were rarely coreferent in the 50 EJV texts. Of the 364 instances that represented pairs of phrases that came from the same sentence, only 26 were positive, i.e., represented coreferent phrases. Nearly half of these (12) were predicate nominative constructions. Most of the remaining positive instances came from uncharacteristically long sentences, all of which had multiple clauses.

  The sentence index is currently computed off-line rather than being explicitly marked by the interface. The sentence boundaries are located, the sentences are numbered, and then an integer (or index) indicating the sentence from which a phrase was extracted was associated with each annotated phrase. A separate interface, for annotating discourse segment, paragraph and sentence boundaries – and perhaps clause boundaries (see below) – may be useful. Such an interface would aid in the construction of additional features for coreference resolution, but would also be useful in testing and developing other language processing components, such as a preprocessor, a noun phrase analyzer, or a dictionary construction tool.

- *Clause Index*

  A frequently occurring pattern in EJV texts is an announcement wherein an organization, *X*, announces that it will take part in some kind of joint venture. One example of this X-SAID-IT pattern is the following:[9]

  ```
  OSAKI ELECTRIC CO., A MANUFACTURER OF POWER DISTRIBUTION
  EQUIPMENT, SAID THURSDAY IT HAS SET UP A JOINT COMPANY IN
  INDONESIA TO PRODUCE INTEGRATING WATT-HOUR METERS.
  ```

  Another example of this announcement phenomenon is illustrated in the following sentence:

[9] See Section 8.2.6.1 for more details on this pattern.

```
NIPPON SANSO K.K. HAS SET UP A JOINT VENTURE WITH A
TAIWANESE INVESTMENT FIRM IN MALAYSIA TO PRODUCE
STAINLESS THERMOS BOTTLES, THE LARGEST JAPANESE OXYGEN
MANUFACTURER SAID FRIDAY.
```

Noting that `OSAKI ELECTRIC CO.,` ... and `IT` are in adjacent clauses of the same sentence, or that `NIPPON SANSO K.K.` and `THE LARGEST JAPANESE OXYGEN MANUFACTURER` are in adjacent clauses in the same sentence, may be useful for coreference resolution, especially if additional information is included about syntactic role (e.g., all four phrases are the subjects of their respective clauses). Unfortunately, this information has not yet been incorporated into the interface.

- *Syntactic Role(s)*

  Knowledge of which syntactic role a phrase plays in a sentence (or clause) is considered useful information for coreference resolution by a number of researchers. For example, many theories rank potential antecedents of an anaphor according to their syntactic role in the previous sentence. Therefore, information about syntactic roles was included in the annotations for the phrases used in the current work.

5.4.3.4 Type Information

Each phrase annotated with CMI must be assigned a high-level category indicating the type of entity to which it refers. For the annotations made in the MUC-5 EJV domain, there were four categories: *organization*, *facility*, *person*, and *industry*. Lower level categories may be assigned via slots associated with each type, e.g., the *relationship* slot associated with *organization* references was used to sub-categorize *organizations* with respect to their status in a tie-up relationship: this slot was used to distinguish joint venture companies from their parent organizations.

5.4.3.5 Slot Information

CMI is a domain-independent tool: a configuration file specifies the different types of entities that are relevant to the domain, as well as the different slots associated with each type and the type of value with which each slot is to be filled. Table 5.1 represents a portion of the configuration file for the MUC-5 EJV domain.

There are four different types of slot fills currently supported by CMI. The following phrase will be used to illustrate each of these slot fill types:

```
ALUMINIUM CO. OF MALAYSIA BHD. (ALCOM), A SUBSIDIARY OF
ALCAN ALUMINUM LTD. OF CANADA
```

- *String:* A substring of the marked string. For example, in the example phrase, the *name* slot would be filled with the substring *ALUMINIUM CO. OF MALAYSIA BHD.* and the *alias* slot would be filled with the substring *ALCOM*.

- *Set:* A prespecified set of possible values. For example, the *type* slot for the example phrase would be filled with the value *company* and the *relationship* slot would be filled with the two values *child* (because it is a `SUBSIDIARY`) and *jv-parent* (because the phrase occurs within the context `VENTURE ... WITH`).

Table 5.1 Types of slot fills supported by CMI

Object Name	Slot Name	Fill Type
organization	name	string
	alias	string
	type	set
	relationship	set
	nationality	normalized
	jv-parent	pointer
	jv-child	pointer
	parent	pointer
	child	pointer
	partner	pointer
facility	name	string
	type	set
	location	normalized
person	name	string
	position	set
	organization	pointer
industry	product/service	string
	type	set
	site	normalized

- *Pointer:* Similar to the *string* slot fill type, except that the string that fills a slot of this type can be part of the surrounding context of the marked string. In the example string, the *parent* slot would be filled with *ALCAN ALUMINUM LTD. OF CANADA*.

- *Normalized:* Also similar to the *string* slot fill type, except that the substring is processed by some normalization procedure, specified elsewhere in the configuration file, to generate a canonical form. The substring `MALAYSIA` in the example phrase would be converted into the canonical form *Malaysia (COUNTRY)* in order to fill the *nationality* slot for that phrase.

In addition to the specifications of entity types and slot fill types, the CMI configuration file also contains a set of heuristics that can be used to propose default slot fill values based on the contents of the marked string. For example, the value *company* will be the default set fill for the *type* slot of any *organization* reference that contains a word or abbreviation commonly used to designate a company, e.g., `COMPANY`, `CORP.` or `INC.`

These heuristics are useful for reducing the annotation effort – it is easier to confirm a preselected default set fill value than to select a value from a list. They may also prove useful for the development and refinement of automated methods for slot fill extraction, i.e., the component of the UMass/Hughes MUC-5 system that was responsible for identifying names, aliases and locations within larger noun phrases, e.g., for the sample sentence above:

```
ALUMINIUM CO. OF MALAYSIA BHD. (ALCOM), A SUBSIDIARY OF
ALCAN ALUMINUM LTD. OF CANADA
```

the slot fill extraction component needed to identify the *name* (`ALUMINUM CO. OF MALAYSIA BHD.`) and *alias* (`ALCOM`). The heuristics have not yet been used for this purpose.

CHAPTER 6

EVALUATING PERFORMANCE

The purpose of the coreference resolution module in an information extraction system is to determine which phrases in a text corefer, so that the representations of all coreferent phrases can be merged together into a single structure. If we are to evaluate the effectiveness of a new approach to coreference resolution, we need to establish some framework in which to compare this approach to other approaches.

Since the coreference resolution task has been posed as a classification problem in this work, the classification *accuracy* of the system might at first seem like the best measure of performance. However, several complications diminish the value of this particular metric in evaluating performance.

This chapter will describe a pair of metrics, *recall* and *precision*, that together provide a more informative evaluation of the performance of a coreference resolution system than accuracy.

6.1 A Simple Approach: Accuracy

An ideal coreference resolution classification system would correctly classify every pair of phrases as coreferent or not coreferent. Unfortunately, coreference classifiers that are designed for any large corpus of texts are likely to make mistakes – even human coreference resolution is prone to errors.

A coreference classifier returns a positive or negative classification for every pair of phrases it is given. Each such classification is either correct or incorrect. A simple measure of the performance of a classifier is the *accuracy* of these classifications, i.e., the ratio of correct classifications to total classifications.[1]

More formally, if we assume that the correct coreference classifications for a given text are listed in a *key* and that the coreference classifications made by some classifier

[1]Conversely, the *error rate* of the classifier can be computed as the ratio of *in*correct classifications to total classifications. Since the error rate is simply the additive inverse of accuracy, either metric provides the same information. We will focus on accuracy, since it will provide an easier comparison point with the other metrics described in this chapter.

Table 6.1 Possible classifications

	Key	
Response	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Table 6.2 Instances representing six relevant references to three entities

<A–B>	<A–C>	<A–D>	<A–E>	<A–F>
	<B–C>	<B–D>	<B–E>	<B–F>
		<C–D>	<C–E>	<C–F>
			<D–E>	<D–F>
				<E–F>

are contained in a system *response*, then the possible outcome of each classification is given in Table 6.1. If we count the number of classifications that fall into each category, then the accuracy of the coreference classifier can be defined as

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + FalsePositive + FalseNegative + TrueNegative}$$

6.1.1 The Problem with Accuracy

We would like a coreference resolution system to be as accurate as possible. However, given a particular level of accuracy, the results of different coreference classifiers can vary widely.

As an example, suppose a text contains references to three relevant entities, and that each entity is referenced twice in the text, for a total of six relevant phrases: we will label these phrases *A, B, C, D, E* and *F*. If we pair each phrase with all the preceding phrases in the text, we get 15 instances, of which three are positive – for simplicity, we will consider *A* and *B*, *C* and *D*, and *E* and *F* to be coreferent. Table 6.2 lists all the instances that would be generated for this text, with the positive instances highlighted in boldface.

Further suppose that we have two different classifiers that are tested on the same 15 instances from this text. *Classifier 1* is a very conservative system, rarely returning a positive classification for a pair of phrases. In this case, it classifies all 15 instances as negative instances of coreference (see Figure 6.1). *Classifier 2*, on the other hand, is more liberal in returning positive classifications; it classifies four of the 15 instances as coreferent — **<A–B>**, <B–C>, <D–E>, and **<E–F>** (see Figure 6.2).

Both classifiers correctly classify 12 of the 15 instances, so they both exhibit a classification accuracy of 80%. However, as can be seen in Figures 6.1 and 6.2, the result of each set of classifications is very different. As a result of the classifications made by *Classifier 1*, a set of six distinct entities is passed on for later discourse processing; the classifications made by *Classifier 2* result in two distinct entities being passed along for further processing.

We need an evaluation framework that captures the differences between these two classifiers. Accuracy is not sufficient for this purpose; a new set of metrics is presented in the next section.

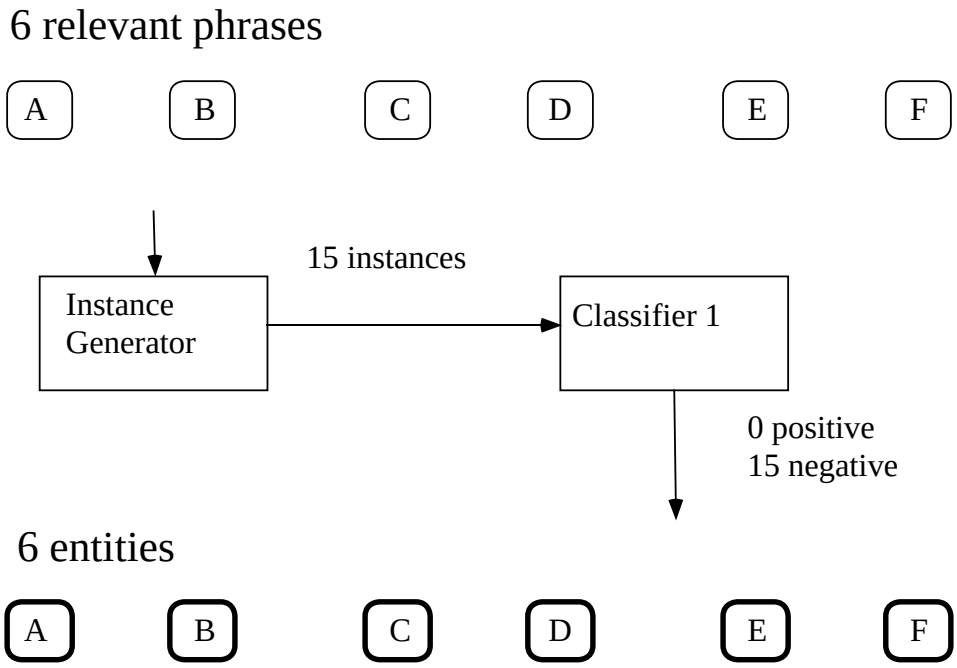


Figure 6.1 Classifier 1

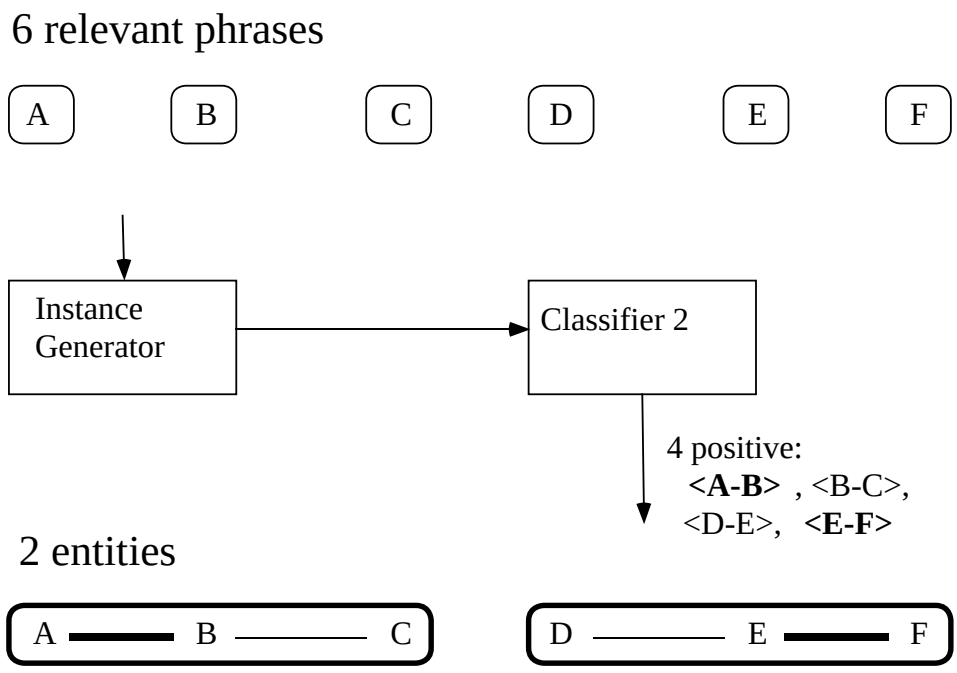


Figure 6.2 Classifier 2

6.2 A More Comprehensive Approach: Recall and Precision

Two metrics that are commonly used in the evaluation of information extraction systems are *recall* and *precision* [Chinchor, 1991].[2] Recall measures the fraction of the information contained in a text that is correctly extracted by a system. Precision measures the fraction of information extracted by a system that is correct.

These two metrics can also be applied to the evaluation of coreference resolution performance:

Recall is the fraction of coreference relationships between phrases in a text that are correctly found by a system.

Precision is the fraction of coreference relationships found by a system that are correct.

Using the categorization of classifications given in Table 6.1, these metrics can be defined more formally as follows:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

If we use these two metrics to evaluate the performance of the two sample classifiers described in the previous section, we can see that these metrics are more effective at illustrating the difference in performance between the two.

There were 15 instances generated for the sample text, of which three were positive: <**A–B**>, <**C–D**>, and <**E–F**>. *Classifier 1* returned negative classifications for all 15 instances, i.e., it found no positive instances. The recall and precision for this classifier can be computed as follows:

$$Recall = \frac{0}{3} = 0\%$$

$$Precision = \frac{0}{0} = ?$$

Classifier 2 returned positive classifications for four of these instances: <**A–B**>, <B–C>, <D–E>, and <**E–F**>. Its recall and precision can be computed as

$$Recall = \frac{2}{3} = 67\%$$

$$Precision = \frac{2}{4} = 50\%$$

Both classifiers exhibit the same classification accuracy, but exhibit very different recall and precision: *Classifier 1* has high precision[3] but low recall; *Classifier 2* has lower precision but higher recall. This tension between recall and precision is well known to researchers in information extraction; its effect on coreference resolution will be discussed in the following section.

[2] These metrics have been prevalent in the *Information Retrieval* community for many years [van Rijsbergen, 1979].

[3] Although precision is undefined for *Classifier 1*, it does not misclassify any positive instances, so under one interpretation, it could be viewed as a high precision classifier.

6.2.1 The Recall/Precision Tradeoff

A coreference classifier that achieves 100% accuracy would also have 100% recall and 100% precision. Few systems, if any, can ever hope to achieve this level of performance on any realistic task domain. The examples in the previous section illustrate the variation in levels of recall and precision that can occur at a given level of accuracy.

There is a fundamental tradeoff between recall and precision. Given two systems that have the same accuracy, the system that is more likely to return positive classifications is more likely to find true coreference relationships among the phrases in a text and thus achieve higher recall; since an imperfect system is also more likely to find coreference relationships among phrases that are not coreferent, precision will suffer. Conversely, a system that is more likely to return negative classifications is less likely to find true coreference relationships among phrases in the text and thus its recall will be lower; however, since such a system is also more likely to find coreference relationships among phrases that are not coreferent, its precision will tend to be higher.

The relative importance between recall and precision is an open question. The effect of the recall/precision tradeoff on overall information extraction performance has yet to be explored.[4] Some researchers working on coreference resolution for information extraction favor high precision over high recall [Appelt *et al.*, 1992, Ayuso *et al.*, 1992]; others favor high recall over high precision [Iwańska *et al.*, 1992].

It may well be the case that the preference for recall or precision may be dictated on a case-by-case basis, depending on the information needs of the system users. It may be the case that the relative importance of these two metrics will vary across different domains.

In any case, the important aspect to note is that a system that maximizes recall often suffers from lower precision, and a system that maximizes precision often suffers from lower recall.

6.2.2 Why not count False Positives and False Negatives?

It may seem that according to this description of the tradeoff between recall and precision, and based on the definitions of recall and precision given in Section 6.2, that one could simply split up the overall error rate and focus on the false negative error rate and false positive error rate instead of recall and precision. Based on the possible classification outcomes listed in Table 6.1, the *overall error rate* can be defined as the inverse of the accuracy metric:

$$\frac{FalsePositive + FalseNegative}{TruePositive + FalsePositive + FalseNegative + TrueNegative}$$

This can be broken down into two parts, where the *false positive error rate* can be defined as

$$\frac{FalsePositive}{TruePositive + FalsePositive + FalseNegative + TrueNegative}$$

and the *false negative error rate* can be defined as

[4] See Section 10.3.5 for additional discussion on future work relating to this topic.

Table 6.3 Classifications on Example Data Set 1

	Key	
Response	Positive	Negative
Positive	40	10
Negative	10	40

Table 6.4 Classifications on Example Data Set 2

	Key	
Response	Positive	Negative
Positive	10	10
Negative	10	70

$$\frac{FalseNegative}{TruePositive + FalsePositive + FalseNegative + TrueNegative}$$

These equations show that recall is inversely correlated with the false negative error rate and precision is inversely correlated with the false positive error rate. While these two error rates do provide more information about the performance of a coreference resolution system than a single accuracy (or error) rate, they are not as useful as recall and precision.

Consider the distribution of classifications given in Tables 6.3 and 6.4. In both of these cases, the false positive error rate and false negative error rate are both 10%. However, the recall and precision for the first data set are both 80%, while the recall and precision for the second data set are both 50%.

The important difference between measuring recall and precision versus measuring the false positive and false negative error rates is that the former pair of metrics depends on the number of actual coreference relationships among phrases in a text and are not directly influenced by the relative distribution of positive and negative instances of coreferent phrases. The false positive and false negative error rates are influenced by both the number of actual coreference relationships and the distribution of positive and negative instances, a distribution that may vary widely across different texts and different domains.

False positive and false negative error rates might be normalized in some way to reduce the variation across texts and domains. However, since recall and precision are used to measure coreference performance in the MUC-6 evaluation (see Section 6.3.1 below), these metrics will be used in evaluating performance throughout this dissertation. The MUC conferences have set the standards for evaluating information extraction systems; the metrics defined and used for the MUC-6 Coreference Task are likely to set the standard for evaluating coreference resolution.[5]

[5] At least with respect to coreference resolution in an information extraction system, which is the primary focus of the work presented in this dissertation.

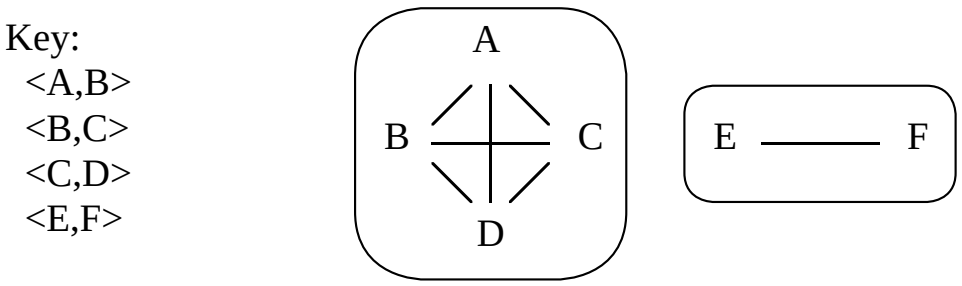


Figure 6.3 Complete graphs representing phrases $\{A,\ B,\ C,\ D\}$ and $\{E,\ F\}$

6.3 A Complication: Transitive Closures

The coreference relation among single-referent phrases[6] is transitive: if phrases A and B are coreferent, and phrases B and C are coreferent, then phrases A and C are also coreferent. Using nodes to represent phrases and links to represent coreferent relationships between pairs of phrases, the transitive closure of a set of coreferent phrases can be represented as a complete graph. Figure 6.3 illustrates an example of transitive closures for two entities: one entity referenced by phrases A, B, C and D, the other entity referenced by phrases E and F.

The existence of transitive closures among coreferent phrases complicates the evaluation of a coreference classifier. While a set of coreferent phrases may entail a complete graph connecting each of the phrases in that set, the same information can be represented more efficiently by a minimal spanning tree of that graph. In the example illustrated by Figure 6.3, one minimal spanning tree for the entity referenced by phrases A, B, C and D could be represented by the links $<A\text{–}B>$, $<B\text{–}C>$ and $<C\text{–}D>$[7]; the minimal spanning tree for the second entity is the same as the complete graph containing the single link for $<E\text{–}F>$. For this example, we will assume that an answer key contains just these four links.

Suppose that a coreference classifier assigns positive classifications to the instances representing the phrase pairs $<A\text{–}B>$, $<A\text{–}C>$, $<D\text{–}E>$ and $<D\text{–}F>$; for representational efficiency, we will assume that these four links appear in the system response. The transitive closures for these coreference links would entail the complete graphs depicted in Figure 6.4.

The simplest way of computing the recall and precision for this system response would be to compare the coreference links in the response with the coreference links in the key directly, i.e.,

$$\begin{aligned} \mathit{Recall} &= \frac{|\textbf{explicit } \mathit{key\ links} \cap \textbf{explicit } \mathit{response\ links}|}{|\textbf{explicit } \mathit{key\ links}|} \\ &= \frac{|\{<A\text{–}B>\}|}{|\{<A\text{–}B>,<B\text{–}C>,<C\text{–}D>,<E\text{–}F>\}|} \\ &= \frac{1}{4} \\ &= 25\% \end{aligned}$$

[6] Section 5.3.2 includes a definition of single-referent and multi-referent phrases.

[7] There are 16 different possible minimal spanning trees for this graph; in general for any set of n coreferent phrases, there exists n^{n-2} distinct minimal spanning trees [Bogart, 1983, pages 134–139].

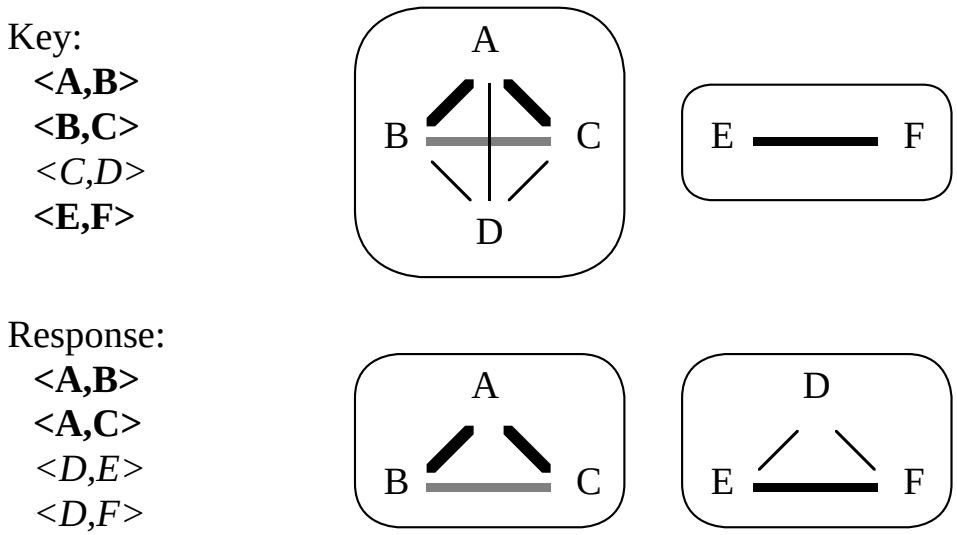


Figure 6.4 Complete graphs representing phrases $\{A,\ B,\ C\}$ and $\{D,\ E,\ F\}$

$$\begin{aligned} Precision &= \frac{|\textbf{explicit } key\ links \cap \textbf{explicit } response\ links|}{|\textbf{explicit } response\ links|} \\ &= \frac{|\{<A\text{–}B>\}|}{|\{<A\text{–}B>,<A\text{–}C>,<D\text{–}E>,<D\text{–}F>\}|} \\ &= \frac{1}{4} \\ &= 25\% \end{aligned}$$

The problem with this simple counting scheme is that it does not give sufficient credit to the system. In particular, the explicit response link <*A*–*C*> which is implicit in the key is not counted, nor is any credit given for the links <*B*–*C*> and <*E*–*F*> which are explicit in the key but implicit in the response.

One solution to this problem is to compare the transitive closure of the key to the transitive closure of the response, i.e.,

$$\begin{aligned} Recall &= \frac{|\textbf{implicit } key\ links \cap \textbf{implicit } response\ links|}{|\textbf{implicit } key\ links|} \\ &= \frac{|\{<A\text{–}B>,<A\text{–}C>,<B\text{–}C>,<E\text{–}F>\}|}{|\{<A\text{–}B>,<A\text{–}C>,<A\text{–}D>,<B\text{–}C>,<B\text{–}D>,<C\text{–}D>,<E\text{–}F>\}|} \\ &= \frac{4}{7} \\ &= 57\% \end{aligned}$$

$$\begin{aligned} Precision &= \frac{|\textbf{implicit } key\ links \cap \textbf{implicit } response\ links|}{|\textbf{implicit } response\ links|} \\ &= \frac{|\{<A\text{–}B>,<A\text{–}C>,<B\text{–}C>,<E\text{–}F>\}|}{|\{<A\text{–}B>,<A\text{–}C>,<B\text{–}C>,<D\text{–}E>,<D\text{–}F>,<E\text{–}F>\}|} \\ &= \frac{4}{6} \\ &= 67\% \end{aligned}$$

The use of transitive closures allows credit to be given for the coreference links that are implicit in both the key and the response. However, in this case, perhaps too much credit is given to the system's performance, i.e., the implicit coreference link <*E*–*F*> is counted toward the precision score, even though this link was implied by two incorrect classifications (<*D*–*E*> and <*D*–*F*>).

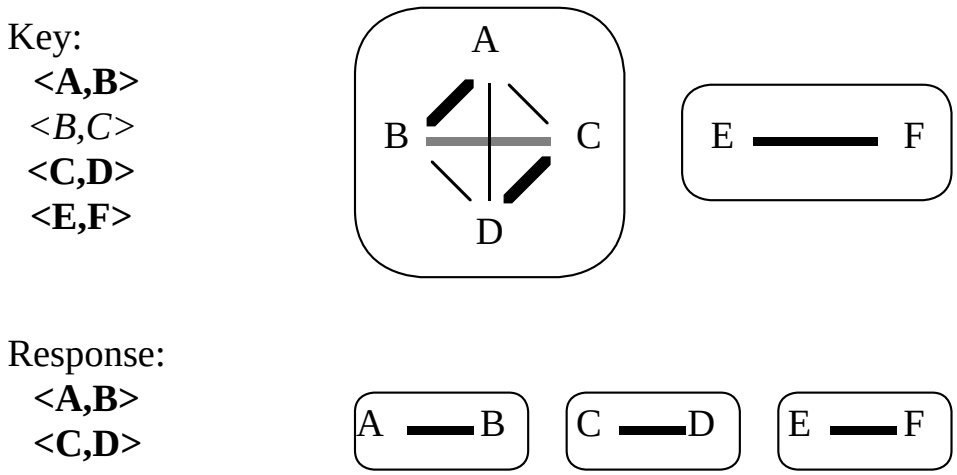


Figure 6.5 Complete graphs representing phrases $\{A,\ B\}$, $\{C,\ D\}$ and $\{E,\ F\}$

Another problem that may result from using *only* the closures of the key and response is that the recall performance of a system may be overpenalized if it misses one explicit link between two subsets of coreferent phrases. To illustrate this potential problem, consider the transitive closures depicted in the graphs shown in Figure 6.5. In this case, the coreference links between phrase pairs $<A–B>$, $<C–D>$ and $<E–F>$ are correctly found by the system. However, the system misses the coreference link $<B–C>$. Comparing the transitive closures for the response and key, we get

$$\begin{aligned} Recall &= \frac{|\textbf{implicit } key\ links \cap \textbf{implicit } response\ links|}{|\textbf{implicit } key\ links|} \\ &= \frac{|\{<A–B>,<C–D>,<E–F>\}|}{|\{<A–B>,<A–C>,<A–D>,<B–C>,<B–D>,<C–D>,<E–F>\}|} \\ &= \frac{3}{7} \\ &= 43\% \end{aligned}$$

$$\begin{aligned} Precision &= \frac{|\textbf{implicit } key\ links \cap \textbf{implicit } response\ links|}{|\textbf{implicit } response\ links|} \\ &= \frac{|\{<A–B>,<C–D>,<E–F>\}|}{|\{<A–B>,<C–D>,<E–F>\}|} \\ &= \frac{3}{3} \\ &= 100\% \end{aligned}$$

6.3.1 The MUC-6 Definitions of Recall and Precision

In order to take maximum advantage of both the explicit and implicit coreference links in the key and the response, we will adopt the scoring scheme proposed for the Coreference Task defined for the Sixth Message Understanding Evaluation and Conference [MUC-6, 1995]:

$$\begin{aligned} Recall &= \frac{|\textbf{explicit } key\ links \cap \textbf{implicit } response\ links|}{|\textbf{explicit } key\ links|} \\ &= \frac{|\{<A–B>,<C–D>,<E–F>\}|}{|\{<A–B>,<B–C>,<C–D>,<E–F>\}|} \\ &= \frac{3}{4} \\ &= 75\% \end{aligned}$$

$$
\begin{aligned}
Precision &= \frac{|\textbf{implicit } \textit{key links} \cap \textbf{explicit } \textit{response links}|}{|\textbf{explicit } \textit{response links}|} \\
&= \frac{|\{<A-B>,<C-D>,<E-F>\}|}{|\{<A-B>,<C-D>,<E-F>\}|} \\
&= \frac{3}{3} \\
&= 100\%
\end{aligned}
$$

For the previous example, illustrated in Figure 6.4, we can recompute the recall and precision scores using the MUC-6 definitions:

$$
\begin{aligned}
Recall &= \frac{|\{<A-B>,<B-C>,<E-F>\}|}{|\{<A-B>,<B-C>,<C-D>,<E-F>\}|} \\
&= \frac{3}{4} \\
&= 75\%
\end{aligned}
$$

$$
\begin{aligned}
Precision &= \frac{|\{<A-B>,<A-C>\}|}{|\{<A-B>,<A-C>,<D-E>,<D-F>\}|} \\
&= \frac{2}{4} \\
&= 50\%
\end{aligned}
$$

The selection of one number (e.g., accuracy) or two numbers (e.g., recall and precision) with which to evaluate a system often results in some loss of information. Even the evaluation of part-of-speech taggers, perhaps the component with the most standardized and widely accepted evaluation framework, is problematic. The standard measure for a part-of-speech tagger is its overall accuracy, i.e., how many part-of-speech labels were correctly assigned by the tagger. Unfortunately, though, all part-of-speech errors are not equally important for most language processing applications; for example, confusing a noun with a verb may be a much more serious error than confusing a singular noun with a plural noun.

The framework defined for evaluating coreference performance for MUC-6 represents a reasonable compromise among competing metrics. The MUC-6 Coreference Task is the first large-scale evaluation of coreference resolution performance; the prominence of a MUC evaluation provides an extra incentive to use the same metrics for evaluating the performance of RESOLVE.

CHAPTER 7

USING DECISION TREES FOR COREFERENCE RESOLUTION

The use of machine learning techniques for coreference resolution may seem like an interesting approach to the problem. However, given the intention of using RESOLVE as a subcomponent of a larger information extraction system, it is important to ascertain whether it can achieve the same level of performance as earlier, manual, approaches.

An early experiment [McCarthy and Lehnert, 1995] shows that a decision tree trained on pairs of coreferent and non-coreferent phrases (positive and negative instances of coreference) can outperform a set of manually engineered rules, where both systems have access to the same knowledge. Some changes have been made to both the format of this experiment and to the underlying data. The details of the revised experiment are the focus of this chapter.

7.1 The Joint Ventures Corpus

The phrases used in all of these experiments were extracted from the MUC-5 English Joint Ventures (EJV) corpus. The articles in the EJV corpus describe business joint ventures among two or more organizations (companies, governments and/or people). The task definition provided for MUC-5 required systems to extract information about the organizations involved, the relationships among these organizations, people affiliated with these organizations, the facilities associated with the joint venture, the products or services offered by the joint venture, its capitalization and revenue projections, and a variety of other related information.

CMI was used to annotate references to organizations, people, facilities, products and services in 50 texts from the MUC-5 EJV corpus. Table 7.1 shows the number of distinct entities (or referents) of each class for which references were annotated, and the total number of references to the set of referents in each class.

Table 7.1 Numbers of distinct referents and references for EJV domain

Class	*# referents*	*# references*
Organization	203	482
Facility	37	65
Person	12	14
Product/Service	48	84
Total	300	645

7.1.1 Organization References from the EJV Corpus

The organization involved in joint ventures were the main focus of most of these articles in the MUC-5 EJV domain, so references to organizations were much more numerous than references to other types of entities, e.g., people. In fact, Table 7.1 shows that over 75% of the relevant references collected from the EJV texts were references to organizations.

The amount of training material influences the accuracy of the concept(s) formed by a machine learning algorithm – generally speaking, performance increases with more training (although the learning curve typically flattens out below 100%).

In order to provide as much training material as possible, *organization* references were selected as the focus of these experiments.

7.1.2 Annotated Phrases

CMI is a graphical user interface that permits the user to mark phrases in a text; for each phrase, the user can indicate the entity(s) with which the phrase is coreferent and some additional information about the phrase that can be inferred either from the phrase itself or its local context. This additional information is parameterized and can be modified easily for use in different domains. The data used in this experiment was based on a set of phrases extracted using CMI.

In principle, much of the information gathered about a particular string could be found automatically: there are numerous proper name recognizer programs, programs that extract location information, and sentence analyzers that can infer relationship information – any system that exhibited good performance in MUC-5 needs to be good at inferring such relationships.

For the purposes of our experiment, however, this information was specified by a user via CMI. The primary motivation for this was to minimize the noise in the data; coreference resolution often occurs at a late processing stage in an information extraction system, and earlier errors such as incorrect part-of-speech tags, incorrectly delimited sentences and semantic tagging errors can create significant noise for a coreference classifier.

CMI was used to mark references to a variety of relevant entity types (*organization*, *facility*, *person* and *product-or-service*) in 50 randomly selected texts.[1] Since references to *organizations* were most numerous, this was the class chosen for the experiment. In the 50 texts, 482 references to a total of 203 *organizations* were marked using CMI.

Some phrases are *multireferent*, i.e., they refer to more than one entity. These multireferent phrases pose difficulties for classification, since it means that some phrases will be coreferent with other phrases in the text that have distinct referents. Thus for a set of phrase pairs which share a given phrase, more than one pair would be classified as a positive instance of coreference. Further complications are created for evaluating the performance of a coreference system when multireferent phrases are included in the data (see Section 6). To simplify the initial experiments reported here, multireferent phrases were excluded from the data set.

[1] In order to make things manageable for CMI annotator, the size of the texts was limited to 2KB, however the majority of texts in the EJV domain fall into this category.

7.2 Manually Engineered Rules vs. Induced Trees

One of the questions that arose early in this work was whether a system that learns how to classify coreferent phrases could achieve the performance of a system that was constructed manually. Machine Learning seemed like an interesting approach to this problem, but was it effective?

An experiment was conducted to compare the performance of the decision trees generated by RESOLVE with the performance of manually engineered rules used for coreference classification in the UMass/Hughes MUC-5 system. The data used in this experiment were based on the MUC-5 EJV data set, described in Chapter 5.

All possible pairings of references from each text were generated, and these pairings were used to create a set of feature vectors used by RESOLVE. The pairings that contained coreferent phrases formed positive instances, while those that contained two non-coreferent phrases formed negative instances.

7.2.1 The MUC-5 Rule-Based Coreference Resolution System

The coreference module of the UMass/Hughes MUC-5 system was designed to minimize false positives, i.e., minimize the likelihood that two phrases that were not coreferent would be labeled coreferent. This design decision was based on the assumption that false positive errors, resulting in the merging of non-coreferent phrases in the final system output, would harm system performance more than false negative errors, which would result in coreferent phrases showing up in distinct structures in the system output representation. This rather conservative approach to coreference was shared by a number of MUC system developers [Appelt *et al.*, 1992, Ayuso *et al.*, 1992], though not by others [Iwańska *et al.*, 1992].

Another factor influencing the coreference module was the short time allotted to developing and testing this system component. Since coreference resolution was a late stage in processing, upstream components had to be stabilized before serious development could take place on coreference. Several late-stage components were being developed in parallel, so it is difficult to assess the time devoted exclusively to developing the coreference module, but we estimate it was two person-weeks.

The rules used to determine whether two phrases (represented as memory tokens) were coreferent in the MUC-5 system are shown in Table 7.2. Following the policy of minimizing false positives, whenever none of the rules fired, the system classified the pair of tokens as not coreferent.

The UMass/Hughes MUC-5 system used a variety of mechanisms to identify phrases referring to joint ventures (the corporate entity formed by two or more parent organizations for some particular business purpose), to identify company names within a phrase (if they exist), and to determine whether one phrase was an alias (an abbreviation or shortened form) of another phrase, as well as the ability to identify trigger families[2] and partitions[3] in the text.

One of the many difficulties in developing the rule set for coreference classification was in ordering the rules. Several different orderings were tested during the development period; this testing was complicated by the fact that the individual rules themselves were being modified concurrently, and the sentence analyzer and other components that were used to generate candidates for coreference resolution were also undergoing development at the same time. The difficulty in rule ordering was

[2] A definition of *trigger family* is provided in Section 7.2.2.1.

[3] A partition is a portion of the text that is focusing on the same main topic. For the MUC-5 system, distinct partitions were recognized only for texts that had bulleted items, as one might see in a news summary of the days headlines. Most texts thus had a single partition.

Table 7.2 The MUC-5 system's coreference rules

IF	both tokens come from the same *trigger family*
THEN	they are not coreferent.
IF	each token comes from a different *partition*
THEN	they are not coreferent.
IF	both tokens contain a common phrase
THEN	they are coreferent.
IF	both tokens refer to joint ventures
THEN	they are coreferent.
IF	both tokens contain the same company name
THEN	they are coreferent.
IF	one token contains an alias of the other
THEN	they are coreferent.
IF	only one token refers to a joint venture
THEN	they are not coreferent.
IF	each token contains different company names
THEN	they are not coreferent.

one of the motivations behind using a machine learning approach – we wanted to develop a system that could *learn* how to combine the positive and negative evidence.

The sequence of rules shown in Table 7.2 was the ordering of the rule set used for final evaluation. This ordering seemed to do a better job than other orderings, but the search for an ordering was not exhaustive, and this final ordering is not assumed to be optimal.

7.2.2 Features Corresponding to MUC-5 Rules

The initial set of features used by RESOLVE was motivated by the antecedents of the rules used in the MUC-5 system coreference module.[4] This set of features, which was used in the experiments reported in this chapter, is shown in Table 7.3; the second column in this table indicates the classification that would have been returned by the rules if the feature in the first column was assigned a positive value. The only MUC-5 rule for which there is no corresponding feature is the second rule, concerning partitions: there were no multi-partition texts among the 50 texts that were annotated for this experiment.

7.2.2.1 SAME-TRIGGER

Do the phrases come from the same trigger family?

Possible values: *YES, NO*

A *trigger word* is a member of a sequence of words that is associated with important domain concepts, e.g., in the phrase *X will form a joint venture with Y*, the

[4] A much larger set of features was used in later experiments.

Table 7.3 Features derived from MUC-5 rules

SAME-TRIGGER	*no*
COMMON-NP	*yes*
BOTH-JV-CHILD	*yes*
SAME-NAME	*yes*
ALIAS	*yes*
XOR-JV-CHILD	*no*
DIFF-NAME	*no*

trigger words might be *form*, *venture*, and/or *with*. A *trigger family* is a set of phrases all off the same trigger word, e.g., a subject and direct object joined by a verb phrase headed by *form*. Being in the same trigger family is essentially equivalent to being in different complement roles of the same verb, and is evidence against the two phrases being coreferent.

7.2.2.2 COMMON-NP

Do the phrases share a common, simple noun phrase?

Possible values: *YES, NO*

Many entity references consist of complex noun phrases, e.g., attached prepositional phrases, relative clauses and appositive constructions. Examples of noun phrases with different levels of complexity include:

- Simple NP: `THE NEW COMPANY`
- Simple NP + PP: `YAKULT HONSHA CO. OF JAPAN` has two constituent simple NPs: `YAKULT HONSHA CO.` and `JAPAN`.
- Appositive: `THE NEW FIRM, P.T. FUJI DHARMA ELECTRIC` has two constituent simple NPs: `THE NEW FIRM` and `P.T. FUJI DHARMA ELECTRIC`.
- Relative Clause: `THE JOINT VENTURE, CALLED P.T. JAYA FUJI LEASING PRATAMA` has two constituent simple NPs: `THE JOINT VENTURE` and `P.T. JAYA FUJI LEASING PRATAMA`.
- Combination: `SUMITOMO, JAPAN'S THIRD LARGEST STEELMAKER BASED IN OSAKA, WESTERN JAPAN` has three constituent simple NPs: `SUMITOMO`, `JAPAN'S THIRD LARGEST STEELMAKER` and `OSAKA, WESTERN JAPAN`.[5]

For example, in text 2348, there is a reference to `THE NEW FIRM, P.T. FUJI DHARMA ELECTRIC` and a later reference to `THE NEW FIRM`; pairing up these phrases would result in a feature value of *YES*.

[5]Location descriptions that include commas were not separated.

Table 7.4 List of phrases associated with references to joint ventures

JOINT
VENTURE
NEW FIRM
NEW COMPANY
TIE-UP
LINKUP
OWNED
PARTNERSHIP
PROJECT

7.2.2.3 JV-CHILD-i

Does phrase i refer to a joint venture company?

Possible values: *YES, NO, UNKNOWN*

A joint venture company (*jv-child*) is a corporate entity that is created as a result of a joint venture between two or more *jv-parent* organizations. Joint venture companies were frequently the primary focus of articles in the EJV domain. Most of these articles only mentioned a single joint venture, so identifying which phrases referred to a joint venture was important for resolving coreference — if both phrases referred to joint ventures, they were likely to be coreferent phrases.

The determination of whether a phrase refers to a joint venture can be made based on either the local context surrounding the phrase or a keyword search of the phrase itself. Most references to joint ventures could be identified by information contained in the phrases themselves. Any phrase that contained one of the substrings listed in Table 7.4 was annotated as a *jv-child*.

Several references to joint ventures, though, could only be identified by the context surrounding the phrases. Some of the patterns that were considered indicative of a reference to a joint venture included:

jv-parent `will` [`form` / `set up` / `establish` / ...] *jv-child*

jv-child [`will be` / `is`] `capitalized at` *money*

jv-child [`will be` / `is`] `owned ... by` *jv-parent*

Examples of these patterns include:

...HIS COMPANY WILL FORM A LOCAL COMPANY WITH JAL ...

THE PARTNERS WILL ESTABLISH
SSB-ANTI CZECHO-SLOVAKIA JAPAN LTD. WITH A CAPITAL OF
300,000 U.S. DOLLARS.

P.T. ORIENTAL SYNTHETIC THREAD WILL BE CAPITALIZED AT 2.7
BILLION RUPIAH[6] ...

DMV SDN. BHD. IS OWNED 55 PCT BY IPOH GARDEN, 40 PCT BY
NISSIN SUGAR AND 5 PCT BY NIPPON STEEL.

The JV-CHILD feature can take on one of three possible values, depending on the phrase itself and its surrounding context:

YES if the phrase refers to a joint venture,

NO if the phrase refers to a *jv-parent* entity[7] and

UNKNOWN otherwise.

7.2.2.4 BOTH-JV-CHILD

Do both phrases refer to a joint ventures?

Possible values: *YES, NO, UNKNOWN*

The BOTH-JV-CHILD feature is defined in terms of the JV-CHILD-i features; it can take on one of three values:

YES if JV-CHILD-1 = *YES* and
JV-CHILD-2 = *YES*,

NO if JV-CHILD-1 = *NO* and
JV-CHILD-2 = *NO*, and

UNKNOWN otherwise

This feature was derived from one of the MUC-5 rules. It is an example of a *meta-feature* in that it is the combination of two more primitive features (JV-CHILD-1 and JV-CHILD-2).

[6]The ***rupiah*** is the basic currency unit of Indonesia.

[7]There were only two examples of an organization that was both the child in one joint venture and the parent in another; since this represented less than one percent of all entities, it seemed reasonable to conclude that if a phrase referenced a ***jv-parent*** entity, it did not also reference a ***jv-child*** entity.

7.2.2.5 XOR-JV-CHILD

Does exactly one phrase refer to a joint venture?

Possible values: *YES, NO, UNKNOWN*

The XOR-JV-CHILD feature is defined in terms of the JV-CHILD-i features; it can take on one of three values:

YES if JV-CHILD-1 = *YES* and
JV-CHILD-2 = *NO*, or
if JV-CHILD-1 = *NO* and
JV-CHILD-2 = *YES*, and

NO if JV-CHILD-1 = *YES* and
JV-CHILD-2 = *YES*,[8] or
if JV-CHILD-1 = *NO* and
JV-CHILD-2 = *NO*, and

UNKNOWN otherwise

Like BOTH-JV-CHILD, this feature is another example of a *meta-feature*, since it is defined in terms of the two more primitive features JV-CHILD-1 and JV-CHILD-2.

7.2.2.6 SAME-NAME

Does each phrase contain exactly the same name?

Possible values: *YES, NO*

If both of the phrases contain names, and those names are the same, this feature has a value of *YES*, otherwise it has a value of *NO*. A *YES* value is strong evidence of a coreferent relationship between the phrases. Note that this feature relies upon a name recognition component that is able to understand, for example, that `FORD MOTOR CO.` is not the name of `FORD MOTOR CO.'S EUROPEAN UNIT`.[9]

7.2.2.7 ALIAS

Is phrase 2 an alias of phrase 1?

Possible values: *YES, NO*

Once a company's full name has been mentioned in a text, subsequent references to that company often contain shortened versions of that name, or aliases. The ALIAS feature looks for substrings in *names* that are found in phrases rather than substrings of the entire phrases. It also considers straightforward acronyms for company names, i.e., those acronyms formed by the first letter in each word comprising the company name (e.g., IBM and International Business Machines Corporation).

[9] A few heuristics that look for possessive constructions and prepositions suffices for most relevant phrases.

An important aspect to note about the ALIAS feature is that it is sometimes true even when the pair of phrases is not coreferent. This happens especially often in the EJV domain because many joint ventures are named after one or more of their parent organizations. Of the 50 texts annotated in this domain, eight references could have been aliases of either a joint venture company or one of the joint venture parent companies. Some examples include:

- `YAKULT HONSHA CO.` (*jv-parent*)
  `PT YAKULT INDONESIA PERSADA` (*jv-child*)
  `YAKULT` (*jv-parent*)

- `FAMILYMART CO.` (*jv-parent*)
  `TAIWAN FAMILYMART CO.` (*jv-child*)
  `FAMILYMART` (*jv-parent*)

- `THE DAIWA BANK` (*jv-parent*)
  `P.T. DAIWA LIPPO LEASING CORP.` (*jv-child*)
  `DAIWA` (*jv-parent*)

One heuristic would have solved this problem for the EJV domain: if a name is a potential alias for a *jv-parent* and a *jv-child*, choose the *jv-parent*. A less domain-specific heuristic would be to choose the shortest name for which a new name is a potential alias.

The computation of features for the experiments reported in this dissertation focused only on a pair of references; it did not take into consideration any other references. Including more context, e.g., the other references encountered in a text, may help improve the accuracy of this feature (its ability to correctly identify the correct full name of a potentially ambiguous alias), and thereby improve performance of the system.

7.2.2.8 DIFF-NAME

Does each phrase contain a different name?

Possible values: *YES, NO*

If both of the phrases contain names, and those names are different, this feature has a value of *YES*, otherwise it has a value of *NO*. A *YES* value is strong evidence of a non-coreferent relationship between the phrases.

7.2.3 Comparing the Two Systems

The features described in the previous section were motivated by the antecedents of the rules used in the UMass/Hughes MUC-5 system's coreference module. In order to compare the performance of the MUC-5 coreference module with the performance of RESOLVE, some sort of conversion would have to take place:

- *Pseudo-Tokens:* The instances generated from the CMI annotations could be converted into the *memory token* format that was used by the MUC-5 coreference module (and other parts of the information extraction system).[10] These

[10]Each memory token contained one noun phrase, one or more lexical patterns encompassing that phrase, part-of-speech tags, semantic features, and information that was inferred from either the

pseudo-tokens could then be passed directly to the rules used in the coreference module, and its output could be used to establish links among the phrases, and these links would form the basis for computing the recall and precision of the system.

- *Pseudo-Rules:* The instances could be used in their current form, but the rules from the MUC-5 coreference module could be represented by a manual combination of the features encoded in the instances. That is, rather than have C4.5 automatically induce a decision tree to classify the instances, a manually constructed decision list, corresponding to the MUC-5 rules (but using the feature definitions used to create the instances), could be used for classification.

The benefit of the pseudo-token approach is that it provides an accurate evaluation of the performance of the coreference module actually used in the UMass/Hughes MUC-5 system. This was the method that was used in an earlier reported experiment [McCarthy and Lehnert, 1995]. The drawback of this approach is that the feature definitions that were used to construct the attribute/value pairs of the instances – which were based on CMI annotations – were different from the code that was used to implement the antecedents of those MUC-5 rules – which were based on CIRCUS output. While the goals of the feature definitions were the same as the goals for code implementing the rule antecedents, there was bound to be some differences in implementation details.

Therefore, a new approach was tried, wherein the exact same feature definitions were used to compare the decision trees automatically induced by C4.5 with a decision list manually constructed to correspond to the MUC-5 rules. While this would no longer provide an accurate evaluation of the *real* MUC-5 coreference module, it would provide a cleaner comparison of the two approaches. This approach was therefore used in the results reported below.[11]

7.2.4 Decision Trees used by RESOLVE

A set of 1660 feature vectors, or instances, was created from the *organization* references marked in the 50 texts. Of these instances, 330 (20%) were *positive* ("+") instances – pairs of phrases that were coreferent – and the remaining 1330 (80%) were *negative* ("-") instances – pairs of phrases that were not coreferent. The distribution of feature values among the 1660 instances is shown in Table 7.5.

Figure 7.1 shows a pruned C4.5 decision tree trained on all the instances.[12] This decision tree can be interpreted as representing a rule such as the one shown in Figure 7.2.

The MUC-5 rules did not make any distinctions between *NO* values and *UNKNOWN* values, i.e., a test either returned a value of *YES* or it did not return a value

phrase or the context in which the phrase was found. This inferred information included the type of entities referenced by the phrase, any name or location substring contained in the phrase, and some domain-specific information such as whether the phrase was a joint venture parent (one of the organizations that formed a joint venture) or joint venture child (the joint venture company itself).

[11] These results differ from those reported by McCarthy and Lehnert [1995]. However, the claim made in that paper, that decision trees can achieve performance at least as good as manually encoded rules, still holds.

[12] Note that for the results presented below in Section 7.2.5, a cross-validation methodology was used, so that the decision trees used for evaluating RESOLVE may not look exactly like this one. The numbers in the parentheses of each leaf node in the decision tree represent the number of training instances represented by that leaf and the number of errors that would be expected when that leaf is used to classify unseen instances.

Table 7.5 Distribution of Values for Features Derived from MUC-5 Rules

	Attribute Values					
	YES		*NO*		*UNKNOWN*	
Attribute Name	+	-	+	-	+	-
JV-CHILD-1	135	341	195	989	0	0
(with unknowns)	135	341	135	735	60	254
JV-CHILD-2	116	353	214	977	0	0
(with unknowns)	116	353	95	576	119	401
SAME-TRIGGER	0	212	330	1118	0	0
COMMON-NP	39	1	291	1329	0	0
BOTH-JV-CHILD	103	9	227	1321	0	0
(with unknowns)	103	9	78	743	149	578
XOR-JV-CHILD	45	676	285	654	0	0
(with unknowns)	2	481	179	271	149	578
SAME-NAME	35	0	295	1330	0	0
DIFF-NAME	75	592	255	738	0	0
ALIAS	106	13	224	1317	0	0

BOTH-JV-CHILD = Y: "+" (113.0/11.7)
BOTH-JV-CHILD = N:
  ALIAS = Y: "+" (119.0/16.1)
  ALIAS = N: "-" (1421.0/125.9)

Figure 7.1 C4.5 decision tree: binary-valued features

IF	BOTH-JV-CHILD = *YES*	THEN	*class* = *coreferent*
ELSE IF	ALIAS = *YES*	THEN	*class* = *coreferent*
ELSE			*class* = *not coreferent*

Figure 7.2 A rule-like representation of the decision tree in Figure 7.1

```
ALIAS = Y: "+" (119.0/16.1)
ALIAS = N:
  BOTH-JV-CHILD = N: "-" (1119.7/104.9)
  BOTH-JV-CHILD = Y:
    XOR-JV-CHILD = Y: "-" (227.2/16.4)
    XOR-JV-CHILD = N:
      DIFF-NAME = Y: "-" (38.5/1.5)
      DIFF-NAME = N: "+" (155.6/47.1)
```

Figure 7.3 C4.5 decision tree: *default* handling of unknown values

of *YES*. Thus two variations were defined for each of the the features JV-CHILD-1, JV-CHILD-2, BOTH-JV-CHILD and XOR-JV-CHILD: in one variation, the features were defined to be binary-valued and all *UNKNOWN* values were converted into *NO* values, i.e., no *UNKNOWN* values were permitted; in the other variation of each feature – which is labeled with "(with unknowns)" in Table 7.5 – *UNKNOWN* values are permitted. The decision tree shown in Figure 7.1 was trained with the binary-valued feature variations. Some of the issues surrounding the use of *UNKNOWN* values are discussed in the next section.

7.2.4.1 The Effect of UNKNOWN Attribute Values

C4.5, as well as most other decision tree induction algorithms, treats attributes with *UNKNOWN* values differently from other values – it selects tests and partitions of the instances based on the set of instances for which an attribute has known values, and then passes *weighted* instances[13] with unknown values down all branches of the tree. Unfortunately, the values of the both the BOTH-JV-CHILD feature and the XOR-JV-CHILD are *UNKNOWN* in many of instances, which can cause C4.5 to split in a rather strange way.

Based on only the known values for the attributes of the BOTH-JV-CHILD and XOR-JV-CHILD features, C4.5 creates the decision tree shown in Figure 7.3, that in some cases tests whether *both* phrases refer to joint venture children, and if they do, it then tests to see if *exactly one* of the phrases refers to a joint venture child! It should be noted that there are *no* instances in the training set for which both these features have positive values, so this is not the result of noise in the data.

Quinlan [1989] catalogs a number of different methods to handling unknown attribute values in three contexts: when evaluating a possible attribute test for a node during decision tree construction, when partitioning the training instances based on a test at a node, and when classifying a new instance. The method he chose for handling unknown values in C4.5 was selected based on its superior performance across a set of seven datasets. While this method may be the best approach in many situations, it may not be the best approach when a large proportion of the instances have unknown values for certain attributes.

When the tree in Figure 7.3 was generated, the meta-features BOTH-JV-CHILD and XOR-JV-CHILD were defined so that the vast majority of instances (73%) had *UNKNOWN* values for these features; the current definitions of these features result

[13] The instance *weight* is based on the distribution of known values for the tested attribute.

```
ALIAS = Y:
  XOR-JV-CHILD = N: "+" (45.0/1.4)
  XOR-JV-CHILD = U: "+" (65.0/7.3)
  XOR-JV-CHILD = Y:
    DIFF-NAME = Y: "-" (8.0/1.3)
    DIFF-NAME = N: "+" (2.0/1.0)
    DIFF-NAME = U: "-" (0.0)
ALIAS = N:
  BOTH-JV-CHILD = N: "-" (795.0/40.8)
  BOTH-JV-CHILD = U: "-" (626.0/88.5)
  BOTH-JV-CHILD = Y:
    DIFF-NAME = Y: "-" (2.0/1.0)
    DIFF-NAME = N: "+" (0.0)
    DIFF-NAME = U: "+" (110.0/9.5)
```

Figure 7.4 C4.5 decision tree: *UNKNOWN* as *first-class* value

in 44% of the instances having *UNKNOWN* values – still a significant portion of the instances. The important aspect to note is that when C4.5 generates decision trees with features for which a large number of instances have *UNKNOWN* values, strange things can occur.[14]

Two possible ways of working around this problem within the context of the C4.5 learning algorithm were considered for the experiments reported in this chapter: disallow unknown values (e.g., lump the unknown values along with the *NO* values), or treat *UNKNOWN* as a *first class* value, the same way that the values *YES* and *NO* are treated.

If *UNKNOWN* values are disallowed, C4.5 generates the decision tree shown in Figure 7.1, which is much more sensible – in that the tree does not contain branches that could never be traversed in classifying actual instances – than the tree depicted in Figure 7.3.

If *UNKNOWN* is treated as a first-class value (like *YES* and *NO*), then C4.5 generates the decision tree shown in Figure 7.4. This tree differs from the decision tree in Figure 7.1 in two important ways: it has swapped the order of tests between ALIAS and BOTH-JV-CHILD[15], and it incorporates the XOR-JV-CHILD into the final tree. As will be shown in Section 7.2.5, this method of handling unknown values results in both improved recall and improved precision.

[14] With the redefined features, a tree was generated in which a subtree with the feature BOTH-JV-CHILD = *YES* had a descendent leaf node with the feature JV-CHILD-1 = *NO*, a combination of features that could not possibly occur in any of the instances.

[15] This is probably due to the effect of having three rather than two possible values for the BOTH-JV-CHILD attribute, which would increase the ***split info*** value, which would decrease the ***gain ratio*** value for this attribute, causing the induction algorithm to select the ALIAS attribute as the best split for the root of the decision tree. See Quinlan [1993], Section 2.2.2, for more details.

```
ALIAS = Y: "+" (120.0/16.1)
ALIAS = N:
 JV-CHILD-1 = N: "-" (1070.0/102.2)
 JV-CHILD-1 = Y:
  JV-CHILD-2 = Y: "+" (112.0/11.7)
  JV-CHILD-2 = N: "-" (351.0/27.0)
```

Figure 7.5 C4.5 decision tree: binary-valued features, no meta-features

7.2.4.2 The Effect of "Meta-features"

The *meta-features* BOTH-JV-CHILD and XOR-JV-CHILD are based on combinations of lower-level features that were not presented to the decision trees shown in Figures 7.1, 7.3 and 7.4:

- JV-CHILD-1, which has the value *YES* when the first phrase refers to a joint venture child, *NO* when the first phrase refers to a joint venture parent, and *UNKNOWN* otherwise.
- JV-CHILD-2, which is defined similarly for the second phrase.

When the meta-features are removed, and replaced with their constituent lower-level features, the concepts represented by the meta-features still show up in the tree (see Figure 7.5). The subtree where JV-CHILD-1 = *YES* and JV-CHILD-2 = *YES* leads to a leaf that returns a positive classification; this subtree represents the BOTH-JV-CHILD concept.

If *UNKNOWN* values are permitted, we get the decision tree shown in Figure 7.6. Under the ALIAS = *YES* branch, the test for XOR-JV-CHILD has been replaced by a partial representation of this concept: if JV-CHILD-1 = *YES* and JV-CHILD-2 = *NO*, then a negative classification is returned; otherwise a positive classification is returned. Note that the other possible set of values that correspond to the XOR-JV-CHILD concept (JV-CHILD-1 = *NO* and JV-CHILD-2 = *YES*) would result in a positive classification.

The structure of decision trees trained without access to meta-features reflects the concepts represented by those meta-features. Not surprisingly, there is no significant difference in the performance between decision trees trained with meta-features available to them and those that are trained without access to meta-features (see Table 7.6).

More will be said on these meta-features in the next chapter, which will focus on the development of a more comprehensive set of features for coreference resolution.

7.2.5 Results

A series of experiments was run using RESOLVE. Each experiment was designed to test a different configuration of RESOLVE, with respect to both its treatment of unknown values and whether or not it used meta-features. A *leave-one-out* cross-validation methodology was used: for each set of instances taken from the 50 texts, one set was selected for testing purposes and the remaining sets were used to train a new decision tree. This process was iterated over all 50 sets of instances.

```
ALIAS = Y:
  JV-CHILD-1 = N: "+" (81.0/5.0)
  JV-CHILD-1 = U: "+" (22.0/4.8)
  JV-CHILD-1 = Y:
    JV-CHILD-2 = Y: "+" (1.0/0.7)
    JV-CHILD-2 = N: "-" (7.0/1.3)
    JV-CHILD-2 = U: "+" (9.0/1.3)
ALIAS = N:
  DIFF-NAME = Y: "-" (592.0/6.2)
  DIFF-NAME = N: "-" (0.0)
  DIFF-NAME = U:
    JV-CHILD-1 = N: "-" (426.0/59.5)
    JV-CHILD-1 = U: "-" (141.0/41.2)
    JV-CHILD-1 = Y:
      JV-CHILD-2 = Y: "+" (110.0/9.5)
      JV-CHILD-2 = N: "-" (162.0/1.4)
      JV-CHILD-2 = U: "-" (102.0/26.6)
```

Figure 7.6 C4.5 decision tree: *UNKNOWN* as *first-class* value, *no* meta-features

The results shown in each row of Table 7.6 represent the average of these iterations. The configurations of RESOLVE are provided in the first two columns of each row – the first column indicates whether *UNKNOWN* values for attributes were considered first-class values as described in Section 7.2.4.1 (indicated by a "Yes" value in column one) or whether *UNKNOWN* values were simply disallowed and treated as *NO* values (indicated by a "No" value in that column); the value in the second column indicates whether or not meta-features, as described in Section 7.2.4.2, were included in the feature set (a "Yes" value in column two indicates that meta-features were *included* in the feature set, a "No" value in that column indicates that meta-features were *excluded*).[16] The last row of the table shows the results from applying the pseudo-rules that represent the coreference rules used in the UMass/Hughes MUC-5 system.

7.2.6 Discussion

The use of meta-features has no significant effect on the performance of RESOLVE, presumably because the lower-level features (JV-CHILD-1 and JV-CHILD-2) that were combined explicitly in the meta-features BOTH-JV-CHILD and XOR-JV-CHILD are automatically combined by C4.5 in similar ways.

The treatment of unknown values appears to have more of an effect on the performance of RESOLVE; however, the differences between recall and precision scores are not statistically significant.[17] The primary difference in the trees generated when *UNKNOWN* is a first-class value is that a subtree appears under the "ALIAS = *NO*" node (see Figure 7.4), where there would otherwise be a leaf (see Figure 7.1). As

[16] Default settings for all C4.5 parameters were used throughout this experiment (see Quinlan [1993], Chapter 9, for more information about C4.5 parameters).

[17] Based on a paired 2-tailed t-test, $p < .05$.

Table 7.6 Results for coreference resolution for EJV *organizations*

System	*Parameters*		*Performance*	
	Unknowns	*Meta-features*	*Recall*	*Precision*
RESOLVE	No	No	64.0%	93.7%
RESOLVE	No	Yes	64.0%	93.7%
RESOLVE	Yes	No	64.9%	94.4%
RESOLVE	Yes	Yes	65.9%	96.2%
MUC-5 rule set	*N/A*		64.0%	93.2%

was noted earlier, the ALIAS feature is imperfect, and is sometimes confused when a joint venture company has a name that closely resembles one or more of its parent companies. The new subtree under the "ALIAS = *NO*" node helps to correct for this confusion: if the second phrase is a potential alias of the first, but the one phrase refers to a joint venture and the other does not, then the phrases are not coreferent. Due to the distribution of *UNKNOWN* values, this important distinction does not arise when *UNKNOWN* values are lumped together with *NO* values.

7.2.7 Conclusions

One of the original goals of this new approach was to develop a system that achieved good performance in resolving references – as good as the performance achieved using manually engineered rules in our MUC-5 system. The results demonstrate that this goal has been accomplished, i.e., the decision trees attain levels of recall and precision that are as high as the levels attained by the coreference resolution rules from the MUC-5 system.

The decision trees, however, achieve this level of performance with less human effort than is required for a manually engineered approach. A knowledge engineer must still define the features, but the C4.5 machine learning algorithm determines how to combine and order these features. One advantage to using a machine learning approach is that it allows a knowledge engineer to focus on determining which features are still needed for resolving references in a particular domain, rather than having to also be concerned with how to combine these features.

These first experiments with applying decision trees to the coreference resolution problem were encouraging. The features used in the experiment described above were not considered comprehensive by any means. While they proved sufficient for attaining a certain level of performance, an examination of specific errors made by the trees shows that additional features were needed to attain higher levels.

The next chapter will describe the features that have since been added, and will show (among other things) that the performance does improve when it is given more knowledge.

CHAPTER 8

THE UTILITY OF DOMAIN-SPECIFIC KNOWLEDGE

Having established the efficacy of using machine learning for coreference resolution, it is now possible to use the machine learning algorithms to explore certain issues relating to the coreference resolution task. In particular, by using ablation experiments, i.e., disabling sets of features, it becomes possible to evaluate the importance of different classes of knowledge to the coreference resolution task.

Domain-specific knowledge, i.e., highly specialized information that pertains only to a narrowly defined topic area, is essential to analyzing texts that focus on a particular topic. For example, at the level of sentence analysis, many different part-of-speech tags and semantic features may be associated with any given word in general; however, when the domain is restricted, the ambiguity is greatly lessened, making sentence processing more reliable.

It is often likewise assumed that domain-specific knowledge is essential for discourse analysis tasks such as coreference resolution. For example, knowledge about which phrases refer to joint venture children and which phrases refer to joint venture parents seems like it would be very important to classifying many pairs of phrases as coreferent or not coreferent. Therefore, this knowledge was encoded in some of the rules used in the MUC-5 system for coreference resolution; Table 8.1 lists the domain-specific rules that were used in the MUC-5 coreference module.

If more time had been available for the development of the MUC-5 coreference module, additional rules would surely had been added, e.g., to look for information about joint venture *parent* companies. The UMass MUC-4 system also contained a number of rules that were based on domain-specific information, such as the ability to distinguish perpetrators of terrorist actions from victims of those actions – identifying phrases as referring to people, or extracting the names of the people or their organizational affiliations, is a domain-independent activity, but the determination of what *role* each person played in a terrorist event required knowledge that was specialized to the domain of Latin American Terrorism.

The focus of this chapter is a set of experiments designed to quantitatively assess the importance of domain-specific features to coreference resolution in the MUC-5 EJV domain. These experiments show that the performance of RESOLVE degrades much more sharply when all of the eight domain-specific features are disabled than when any other set of eight domain-independent features are disabled.

Table 8.1 Domain-specific rules used in the MUC-5 system

IF	both tokens refer to joint ventures
THEN	they are coreferent.
IF	only one token refers to a joint venture
THEN	they are not coreferent.

The first section of this chapter defines the notion of domain-independence used to partition the features. The second section defines the features in the domain-independent partition. The domain-specific features are defined in the third section below. The experiment is described in the fourth section. The last two sections discuss two questions that arise in these experiments: why is RESOLVE not achieving 100% recall even when it is using all of its features, and why does performance go down so dramatically when the domain-specific features are disabled.

8.1 Domain-Specific vs. Domain-Independent Features

One way to categorize features used in coreference resolution is along the dimension of domain dependence, i.e., how much a feature depends on knowledge that is specific to a particular domain.

Domain dependence can be a difficult concept to specify. What if a feature is common to a set of domains, rather than a single, narrowly defined domain? What if the feature is common to many domains, but some aspects of the feature definition might be specially tailored to different domains?

The interpretation of domain dependence taken in the present work is based on its task orientation – information extraction. Any linguistic processing that is needed for a variety of domains is likely to be included in the domain-independent portion of a system that is used in every application of the system to a new domain. Thus, any feature that is common to a set of domains will be counted as a domain-independent feature.

The full set of features used in the experiments reported in this chapter is listed in Table 8.2; this table also includes the distribution of values for these features among all of the instances used for training and testing.

8.2 Domain-Independent Features

Many of the features that are useful for coreference resolution are common to a variety of domains. Sources of knowledge for such features include part-of-speech tags, syntactic and semantic analysis and simple string comparisons. Some features require special-purpose pattern matching systems, such as a proper name recognizer, but these systems work across domains and are not specific to one particular domain.

8.2.1 Features based on Keywords

Some of the features used by RESOLVE can be extracted from phrases based on a keyword analysis of those phrases. Each of these features is computed by searching a phrase for an enumerated list of words; if the word is found (possibly only in a particular position), the feature is assigned a value of *YES*, else it is is assigned a value of *NO*.

8.2.1.1 DEF-ART-i

Does phrase i start with a definite article?

Possible values: *YES, NO*

Table 8.2 Distribution of Feature Values for MUC-5 EJV Domain

	Attribute Values					
	YES		*NO*		*UNKNOWN*	
Attribute Name	+	-	+	-	+	-
DEF-ART-1	72	233	274	1139	0	0
INDEF-ART-1	86	260	260	1112	0	0
PRONOUN-1	15	47	331	1325	0	0
LOC-1	100	440	246	932	0	0
NAME-1	198	912	148	460	0	0
GOVERNMENT-1	2	19	278	1060	66	293
JV-PARENT-1	132	725	137	385	77	262
JV-CHILD-1	136	344	137	747	73	281
DEF-ART-2	134	373	212	999	0	0
INDEF-ART-2	24	160	322	1212	0	0
PRONOUN-2	19	21	327	1351	0	0
LOC-2	75	399	271	973	0	0
NAME-2	188	907	158	465	0	0
GOVERNMENT-2	2	8	196	926	148	438
JV-PARENT-2	101	586	121	415	124	371
JV-CHILD-2	118	376	109	609	119	387
SAME-TRIGGER	0	212	346	1160	0	0
SAME-SENTENCE	32	401	314	971	0	0
PREVIOUS-SENTENCE	123	397	223	975	0	0
SAME-CONSTITUENT	163	524	183	848	0	0
BOTH-SUBJECT	125	286	221	1086	0	0
SAME-STRING	17	3	329	1369	0	0
SUB-STRING	114	13	232	1359	0	0
COMMON-NOUN	89	57	257	1315	0	0
COMMON-NM	95	91	251	1281	0	0
COMMON-NM/NOUN	187	170	159	1202	0	0
COMMON-NP	39	3	307	1369	0	0
COMMON-LOC	20	25	0	95	326	1252
BOTH-JV-PARENT	71	257	108	558	167	557
BOTH-JV-CHILD	104	9	83	770	159	593
XOR-JV-PARENT	3	521	176	294	167	557
XOR-JV-CHILD	3	498	184	281	159	593
BOTH-GOVERNMENT	2	0	162	717	182	655
SAME-NAME	35	0	76	601	235	771
DIFF-NAME	76	601	35	0	235	771
ALIAS	107	13	239	1359	0	0
X-SAID-IT	7	0	339	1372	0	0
X-IS-Y	14	0	332	1372	0	0

Most phrases that start with a definite article are definite anaphoric references, and should be resolved with an earlier reference. Note that definite articles are not always indicative of anaphoric reference: some organization names include `THE` in their titles, e.g., `THE FUJI BANK` or `THE SALIM GROUP`; references to governments often begin with definite articles, e.g., `THE INDONESIAN GOVERNMENT`; references that include a comparative modifier, e.g., `THE THIRD LARGEST BRAZILIAN LIME MAKER`.

8.2.1.2 INDEF-ART-i

Does phrase i start with an indefinite article?

Possible values: *YES, NO*

Indefinite articles usually introduce new entities into a discourse, so references that start with `A` or `AN` normally should not be resolved with any previous references. As with definite articles, there are exceptions. For example, in predicate nominative expressions, the second reference often starts with an indefinite article, e.g.,

```
TRANS-MEDIA RESOURCES IS A TOKYO-BASED BUSINESS CONSULTING
FIRM
```

Some phrases are intended to include a previously mentioned entity in a class of entities, e.g., in a reference to `FAMILYMART CO.`, which is opening a convenience store, `FAMILYMART` is included in the class of Japanese convenience store operators.

```
THIS WILL BE THE FIRST OVERSEAS STORE TO BE RUN BY A
JAPANESE CONVENIENCE CHAIN STORE OPERATOR.
```

8.2.1.3 PRONOUN-i

Is phrase i a pronominal reference?

Possible values: *YES, NO*

For references to single EJV *organizations*[1], this feature had a value of *YES* only when the phrase consisted of the single word `IT`. Since only entire noun phrases are considered for coreference candidates this feature does not apply to phrases that contain pronominal substrings as part of a larger string, e.g., possessive pronouns such as `HIS` in `HIS COMPANY`.

8.2.1.4 GOVERNMENT-i

Does phrase i refer to a government entity?

Possible values: *YES, NO*

[1]Multi-referential phrases were excluded, as was the case in the experiments reported in Chapter 7; cf. Section 5.3.2.

Table 8.3 List of *generic organization descriptor* strings

COMPANY
CONCERN
FIRM
MAKER
MAKERS
GOVERNMENTS
PARTNERS
WORKS
ENGINEERING

There are very few relevant references to government entities in the EJV domain – the overwhelming majority of entities involved in joint ventures are companies of some kind. However, when one government entity was involved in a joint venture, there was never a second government entity directly involved in that venture. Thus, this feature was added to help isolate the references to government entities – two phrases referring to government entities were likely to be coreferent, and any phrase referring to a government was unlikely to be coreferent with another phrase that did not refer to a government.

This feature was defined to be true for any phrase that contained the word `GOVERNMENT` or `STATE` (unless it was `STATE-OWNED`), and for any phrase that contained only a country name.

8.2.1.5 BOTH-GOVERNMENT

Do both phrases refer to government entity?

Possible values: *YES, NO, UNKNOWN*

When both phrases of a pair refer to government entities, they are likely to be referring to the same government entity. This feature was added to help identify such cases. The value of the BOTH-GOVERNMENT feature is defined as follows:

YES if GOVERNMENT-1 = *YES* and
GOVERNMENT-2 = *YES*

NO if both phrases can be identified as corporate entities by the presence of generic organization descriptors such as `COMPANY` or `FIRM` (see Table 8.3) or corporate designators such as `CORP` or `INC` (see Table 8.4).

UNKNOWN otherwise

In other domains, there may well be relevant references to many different government entities, so this feature may be less useful; however, discriminating government entities from corporate entities — via the GOVERNMENT-i features — is likely to be a useful distinction in many domains.

Table 8.4 List of *corporate designator abbreviation* strings

B.H.D.
CO.
CORP.
LTD.
INC.
P.T.
K.K.
S.A.
S.D.N.

8.2.2 Features Based on String Matching

Two features were used in order to capture some rather straightforward relationships that may have otherwise been missed by the other features – whether two phrases are identical or whether one phrase is a sub-string of the other.

8.2.2.1 SAME-STRING

Are the phrases identical?

Possible values: *YES, NO*

The only relevant references to *organizations* that were identical and *not* coreferent were some pronominal references (e.g., `IT`), although some identical pronominal references *were* coreferent. Over half (12) of the (20) identical phrases were names, or aliases of the names, of an organization; several (5) were definite references to a joint venture company, e.g., `THE VENTURE`; the remaining three, all of which were negative instances, were pronominal references.

8.2.2.2 SUB-STRING

Is phrase 2 a substring of phrase 1?

Possible values: *YES, NO*

Leading articles are deleted, so that, for example,
`THE JOINT VENTURE`
would be a substring of
`A JOINT VENTURE IN MALAYSIA`
The SUB-STRING feature is not symmetric, i.e., it does not check whether the first phrase is a substring of the second phrase. This is because subsequent references to entities are often shortened forms of earlier references; if the new phrase is longer and the older phrase is a substring of it, it may well be that the new phrase is introducing a new entity. In the training data, a new phrase being a sub-string of an older phrase was more highly correlated with coreference than the older phrase being a sub-string of a new phrase (90% versus 77%).

The SUB-STRING feature is similar to the ALIAS feature: the ALIAS feature is restricted in that it looks only at the *name* fields of phrases rather than the entire phrases, but it is broadened in that it considers straightforward acronyms for company names.

Two examples will help illustrate how these features differ. In the first example, the ALIAS feature is true — the *name* field of the second phrase is a substring of the *name* field of the first phrase — but the SUB-STRING feature is false — since the second phrase is actually longer than the first phrase, it could not possibly be a sub-string of it.

1. `MITSUBISHI MINING AND CEMENT CO.`

2. `MITSUBISHI, THE FOURTH LARGEST JAPANESE CEMENT PRODUCER`

There are many cases where one phrase is a sub-string of another, but one or both phrases do not contain any name information, e.g.,

1. `THE NEW VENTURE, HUNI FERMENTATION LTD.`

2. `THE VENTURE`[2]

8.2.3 Features Based on Proper Name Recognition

Information about proper names can be very useful for coreference resolution. The features described in this section are based on the ability to recognize a name, e.g., a person's name or the name of a company, or a location, e.g., the name of a city or country, when they occur within a phrase.

8.2.3.1 NAME-i

Does phrase i contain a name?

For EJV *organization* references, this feature encompasses both the full *name* of an *organization* as well as any shortened forms, or *aliases*, of an *organization*.

The presence or absence of a name in a phrase is useful information when combined with other features: if both phrases contain *name* fields, and one *name* is not an alias (see Section 8.2.3.4) of the other *name*, then the two phrases are probably not coreferent.

8.2.3.2 SAME-NAME

Does each phrase contain exactly the same name?

Possible values: *YES, NO*

If both of the phrases contain names, and those names are the same, this feature has a value of *YES*, otherwise it has a value of *NO*.[3]

[2] Note that the SUB-STRING feature ignores leading articles.

[3] This feature was described in much more detail in Section 7.2.2.6. There were no changes made to the definition of this feature for the experiments reported in this chapter.

8.2.3.3 DIFF-NAME

Does each phrase contain a different name?

Possible values: *YES, NO*

If both of the phrases contain names, and those names are different, this feature has a value of *YES*, otherwise it has a value of *NO*.[4]

8.2.3.4 ALIAS

Is phrase 2 an alias of phrase 1?

Possible values: *YES, NO*

The ALIAS feature looks for substrings in the *name* fields of phrases rather than substrings of the entire phrases. It also considers straightforward acronyms for company names, i.e., those acronyms formed by the first letter in each word comprising the company name (e.g., IBM and International Business Machines Corporation). If both phrases contain names, and the second phrase contains a name that is a substring or acronym of the name in the first phrase, then the value of this feature is *YES*, otherwise it is *NO*.[5]

8.2.3.5 LOC-i

Does phrase i contain any location information?

Possible values: *YES, NO*

Such information is often found in prepositional phrases, possessive constructions and various modifiers of the phrases, e.g.,

- `A JOINT VENTURE IN MALAYSIA`
- `TAIWAN'S LARGEST CAR DEALER`
- `THE INDONESIAN INDUSTRIAL GIANT, THE SALIM GROUP`
- `TOKYO-BASED NIPPON SANSO`

8.2.3.6 COMMON-LOC

Do the phrases have compatible location information?

Possible values: *YES, NO, UNKNOWN*

The value of this feature is true if two references have the exact same location, or if one location is a more specific location than the other, e.g.,

`TOKYO-BASED NIPPON SANSO`

`THE LARGEST JAPANESE OXYGEN MANUFACTURER`

[4]This feature was described in much more detail in Section 7.2.2.8. There were no changes made to the definition of this feature for the experiments reported in this chapter.

[5]This feature was described in much more detail in Section 7.2.2.7. There were no changes made to the definition of this feature for the experiments reported in this chapter.

8.2.4 Features Based on Syntactic Analysis

The features used in an early experiment [McCarthy and Lehnert, 1995] were based entirely on semantic information – no syntactic information was available to RESOLVE at that time. Since then, some information has been added to the annotations: constituent buffer information (subject, direct object, prepositional phrase) and sentence index within a text. This information is used to compute the features listed in this section.

8.2.4.1 SAME-TRIGGER

Do the phrases come from the same trigger family?

Possible values: *YES, NO*

As was mentioned in Section 7.2.2.1, Being in the same trigger family is essentially equivalent to being in different complement roles of the same verb, and is evidence against the two phrases being coreferent.

8.2.4.2 SAME-SENTENCE

Do the two phrases occur in the same sentence?

Possible values: *YES, NO*

Most *relevant* referents are mentioned only once in any given sentence. There are some special cases of multiple references to a referent within a single sentence; two of these general cases are handled by the specially designed features X-SAID-IT and X-IS-Y (see sections 8.2.6.1 and 8.2.6.2 below), which together account for most of these cases.

8.2.4.3 PREVIOUS-SENTENCE

Do the two phrases occur in adjacent sentences?

Possible values: *YES, NO*

Many coreference resolution theories and algorithms place a special emphasis on the concept of recency, i.e., the distance between an anaphor and its antecedent [Winograd, 1972]. Usually, more recent (closer) phrases are preferred as antecedents over less recent (farther) phrases. Some theories have gone so far as to preclude resolving an anaphor with any phrase more than one sentence away [Brennan *et al.*, 1987, Hobbs, 1978].

This feature was added in order to capture some of this recency information.

8.2.4.4 BOTH-SUBJECT

Are both phrases subjects in their respective clauses?

Possible values: *YES, NO*

One of the observations made during the annotation of the 50 EJV texts is that pronouns and definite noun phrases that occur in the subject of one sentence are often coreferent with the subject of the previous sentence. This feature was added in the hope that it would be combined with the PREVIOUS-SENTENCE feature by the learning algorithm.

8.2.4.5 SAME-CONSTITUENT

Do the two phrases occur in the same constituent?

Possible values: *YES, NO*

This feature represents a broadening of the BOTH-SUBJECT feature; for example, it is assigned a value of *YES* if both phrases are subjects or both are direct objects. If the phrases are used in different syntactic roles of their respective sentences, the feature is assigned the value *NO*.

This feature is based on the observation that different sentences within a text often follow a similar pattern with respect to the order in which they refer to the relevant entities. For example, in the first sentence of one text, we have:

```
SUMITOMO ELECTRICAL INDUSTRIES LTD. SAID MONDAY IT SIGNED A
CONTRACT WITH AN INDONESIAN CONGLOMERATE TO FORM A JOINT
VENTURE TO MANUFACTURE MATERIALS USED FOR CIVIL ENGINEERING
PROJECTS.
```

In the third sentence of that text, we have

```
SUMITOMO ELECTRICAN, JAPAN'S LARGEST MAKER OF ELECTRICAL
WIRES AND CABLES, WILL SET UP THE VENTURE WITH THE
INDONESIAN INDUSTRIAL GIANT, THE SALIM GROUP, AND SUMITOMO
CORP., A LEADING TRADING FIRM.
```

The subjects of each of these two sentences are coreferent;[6] however, the prepositional phrases starting with `WITH` and attached to the direct objects `CONTRACT` and `VENTURE` are also coreferent, i.e., `AN INDONESIAN CONGLOMERATE` and `THE INDUSTRIAL GIANT, THE SALIM GROUP`.

[6]The latter sentence contains a typographical error; while this is not much of a problem for a human reader, it creates significant problems for a computerized coreference resolution system.

8.2.5 Features Based on Noun Phrase Analysis

The general form of a noun phrase (NP), whose definition is strongly influenced by the CIRCUS sentence analyzer, was discussed in Sections 3.2.1. The general format is duplicated below, for easy reference:

$$
\begin{array}{lcl}
\textit{SimpleNP} & = & [< \textit{article} >][< \textit{noun-modifier} >]* < \textit{head-noun} > \\
NP & = & \textit{SimpleNP} \\
NP & = & NP < \textit{preposition} > NP \\
NP & = & NP(NP) \\
NP & = & NP < \textit{past participle verb phrase} > NP \\
NP & = & NP, NP
\end{array}
$$

The features in this section are based on an analysis of noun phrases that is able to separate complex noun phrases into their constituent simple noun phrases, in order to identify which words are modifiers and which are head-nouns.

8.2.5.1 COMMON-HEAD-NOUN

Do the phrases share a common head noun?

Possible values: *YES, NO*

An annotated phrase may contain a number of constituent simple NPs, each of which ends with a head noun. Since many phrases referenced parts of a larger whole, and it would be easy for a learning algorithm to confuse the part with the whole, head nouns of attached prepositional phrases were not included in the computation of this feature. For example, `TOYO REAL ESTATE CO., A SUBSIDIARY OF SANWA BANK` and `SANWA BANK` have a common head noun (and a common modifier), but a learning algorithm should not conclude that `SANWA BANK` is the same as `A SUBSIDIARY OF SANWA BANK`); therefore, the head nouns would be `CO.` and `SUBSIDIARY` but not `BANK`.

8.2.5.2 COMMON-MODIFIER

Do the phrases share a common modifier?

Possible values: *YES, NO*

Given the general form of an NP, this feature checks for matches among all the modifiers, i.e., all the words between the article and head noun, of the constituent simple NPs of each of the phrases. As with the COMMON-HEAD-NOUN feature, attached prepositional phrases are excluded from this matching process.

8.2.5.3 COMMON-HEAD-NOUN/MODIFIER

Do the phrases share a common head noun ***or*** *modifier?*

Possible values: *YES, NO*

Since according to the NP definition used in the present work, there is only a single head noun per simple NP, and this head noun is often dropped in subsequent references, this feature was designed to capture similarities among shortened forms of phrases. For example, `GENERAL MOTORS CORP.` and `GENERAL MOTORS` do not share common head nouns, but they do share a common modifier (`GENERAL`) and a second modifier from the first phrase (`MOTORS`) is the head noun of the second phrase.

8.2.5.4 COMMON-NP

Do the phrases share a common, simple noun phrase?

Possible values: *YES, NO*

Many entity references are complex noun phrases, e.g., attached prepositional phrases, relative clauses and appositive constructions (see Section 3.2.1). This feature is assigned a value of *YES* whenever two phrases have a simple noun phrase, that may be a constituent of a more complex noun phrase, in common; otherwise it is assigned the value *NO*. The following is an example of a pair of phrases for which COMMON-NP = *YES*:
`THE NEW FIRM, P.T. FUJI DHARMA ELECTRIC` and `THE NEW FIRM`

8.2.6 Special-Purpose Features

Some special patterns were observed during the annotation of the training data for which special features were constructed. These patterns include announcements made by some entity and predicate nominative constructions.

8.2.6.1 X-SAID-IT

Do the phrases fit the pattern X-SAID-IT?

Possible values: *YES, NO*

A commonly occurring pattern in EJV texts was some organization announcing that it was forming a joint venture with one or more other organizations; this pattern had the general form of

"*organization-name communication-verb* [*date*] [`that`] `it` ..."

The following example illustrates this pattern.

> SUMITOMO ELECTRICAL INDUSTRIES LTD. SAID MONDAY IT SIGNED A CONTRACT WITH AN INDONESIAN CONGLOMERATE TO FORM A JOINT VENTURE TO MANUFACTURE MATERIALS USED FOR CIVIL ENGINEERING PROJECTS.

Linking the *organization-name* with the pronoun `IT`, the subject of the clause about forming the venture, was very important in the EJV domain; without such a link, the system might miss the *organization name*, since patterns involving communication verbs are not highly correlated with relevant references.[7]

A special heuristic for this local coreference resolution problem was added to the version of the CIRCUS sentence analyzer used in MUC-5. CIRCUS would carry over the *organization-name* from the subject buffer of the first clause (governed by the communication verb) into the subject buffer of the second clause (governed by the creation verb phrase), overwriting `IT`.

Since RESOLVE was designed to be independent of any particular sentence analyzer, it cannot rely on coreference resolution of examples of this pattern to be done

[7]Lots of organizations and people issue statements in news articles, but some of these organizations and people are not relevant to the IE task, and many of the things they say are not relevant.

during sentence analysis. It thus includes a special feature created in order to resolve this pattern.

This feature is important, not only for the purposes of correctly resolving occurrences of this pattern, but also to interact with the SAME-SENTENCE feature: SAME-SENTENCE is true in all instances for which X-SAID-IT is true. If the X-SAID-IT can be used to partition the instances for which SAME-SENTENCE is true, then a smaller proportion of those instances will be positive, and the SAME-SENTENCE feature is more likely to be indicative of non-coreferent phrases.

8.2.6.2 X-IS-Y

Do the phrases fit the pattern X-IS-Y?

Possible values: *YES, NO*

Predicate nominatives constitute another common pattern among relevant clauses in EJV texts:

"*entity-reference-1 to-be-verb entity-reference-2*"

In many cases, either the subject or the direct object contained the name of the *organization*, and the other constituent contained additional information about it. Examples include:

THE TAIWANESE FIRM IS SANWU BANDO INC.

THE FUJI BANK WILL BECOME THE SECOND JAPANESE CITY BANK TO ENGAGE IN LEASING BUSINESS IN THIS COUNTRY

This feature, like X-SAID-IT, may be important for splitting off a subset of the instances for which SAME-SENTENCE is true, leaving behind a smaller proportion of positive instances, further increasing the likelihood that SAME-SENTENCE will help identify negative instances of coreference.

8.3 Domain-Specific Features

Some features used for classifying coreferent phrases are particular to a domain. For example, in the EJV domain, distinguishing which phrases refer to entities that are formed as a result of a business joint venture — the *jv-child* entities — from phrases that refer to the organizations that formed the joint venture — the *jv-parent* entities — is very useful.

Another category of features used in coreference resolution are based on knowledge that is useful in a variety of domains, but may differ slightly from domain to domain, or among different types of entities. For example, many entities have names, and there are certain broad naming conventions that are applied to most entities; however, a name recognizer may require some special tuning to pick out names of specific types of entities in specific domains.

8.3.1 Simple Features Based on a Single Phrase

One set of domain-specific features is based on individual phrases, e.g., is a phrase referring to a joint venture company, or to the parent company of a joint venture? These features are computed from either the contents of the individual phrases (the words making up the phrases) or their surrounding context.

8.3.1.1 JV-PARENT-i

Does phrase i refer to a joint venture parent organization?

Possible values: *YES, NO, UNKNOWN*

A joint venture parent (*jv-parent*) is an organization that has joined with one or more other organizations to create a new corporate entity or to pursue joint work on some project. The determination that some phrase refers to a *jv-parent* organization is usually based on the context in which that phrase occurs. Examples of context in which a phrase, X_i, likely refers to a *jv-parent* organization include

- "X_1 **formed a venture with** X_2"
- "**venture between** X_1 **and** X_2"
- "**the venture will be owned 65% by** X_1 **and 35% by** X_2"

Since *jv-parent* organizations are rarely themselves *jv-child* organizations, this feature has three possible values, based on the phrase itself and its surrounding context:

YES if the phrase refers to a *jv-parent*

NO if the phrase refers to a *jv-child*, and

UNKNOWN otherwise.

8.3.1.2 JV-CHILD-i

Does phrase i refer to a joint venture company?

Possible values: *YES, NO, UNKNOWN*

The JV-CHILD feature can take on one of three possible values, depending on the phrase itself and its surrounding context:[8]

YES if the phrase refers to a *jv-child*,

NO if the phrase refers to a *jv-parent*, and

UNKNOWN otherwise.

[8]This feature was described in much more detail in Section 7.2.2.3. There were no changes made to the definition of this feature for the experiments reported in this chapter.

8.3.2 Meta-Features Based on a Pair of Phrases

Some individual features can be combined in ways that are potentially very useful for coreference resolution. However, machine learning algorithms may not find such combinations, or, in the case of a decision tree induction algorithm such as C4.5, the individual features may be separated by many levels in a decision tree.

The FRINGE algorithm [Pagallo, 1989] was developed to enable decision trees combine individual features into Disjunctive Normal Form (DNF) expressions, an approach that has been shown to improve classification accuracy on several data sets. The approach taken in this work, however, is to use domain knowledge to manually combine those individual features in ways that appear useful to a knowledge engineer.

In order to ensure that certain combinations of features are found by the machine learning algorithm, they can be explicitly combined into *meta-features*. The domain-specific meta-features created for the MUC-5 EJV domain are described in this section.

8.3.2.1 BOTH-JV-PARENT

Do both phrases refer to joint venture parent organizations?

Possible values: *YES, NO, UNKNOWN*

The BOTH-JV-PARENT feature is defined in terms of the JV-PARENT-i features; it can take on one of three values:

YES if JV-PARENT-1 = *YES* and
JV-PARENT-2 = *YES*

NO if JV-PARENT-1 = *NO* and
JV-PARENT-2 = *NO*

UNKNOWN otherwise

8.3.2.2 XOR-JV-PARENT

Does exactly one phrase refer to a joint venture parent organization?

Possible values: *YES, NO, UNKNOWN*

The XOR-JV-PARENT feature is defined in terms of the JV-PARENT-i features; it can take on one of three values:

YES if JV-PARENT-1 = *YES* and
JV-PARENT-2 = *NO*, or
if JV-PARENT-1 = *NO* and
JV-PARENT-2 = *YES*

NO if JV-PARENT-1 = *YES* and
JV-PARENT-2 = *YES*, or
if JV-PARENT-1 = *NO* and
JV-PARENT-2 = *NO*

UNKNOWN otherwise

8.3.2.3 BOTH-JV-CHILD

Do both phrases refer to joint venture companies?

Possible values: *YES, NO, UNKNOWN*

The BOTH-JV-CHILD feature is defined in terms of the JV-CHILD-i features, in a way that parallels the definition of BOTH-JV-PARENT; it can take on one of three values:[9]

YES if JV-CHILD-1 = *YES* and
JV-CHILD-2 = *YES*,

NO if JV-CHILD-1 = *NO* and
JV-CHILD-2 = *NO*, and

UNKNOWN otherwise

8.3.2.4 XOR-JV-CHILD

Does exactly one phrase refer to a joint venture company?

Possible values: *YES, NO*

The XOR-JV-CHILD feature is defined in terms of the JV-CHILD-i features, in a way that parallels the definition of XOR-JV-PARENT; it can take on one of three values:[10]

YES if JV-CHILD-1 = *YES* and
JV-CHILD-2 = *NO*, or
if JV-CHILD-1 = *NO* and
JV-CHILD-2 = *YES*, and

NO if JV-CHILD-1 = *YES* and
JV-CHILD-2 = *YES*, or
if JV-CHILD-1 = *NO* and
JV-CHILD-2 = *NO*, and

UNKNOWN otherwise

[9] This feature was described in much more detail in Section 7.2.2.4. There were no changes made to the definition of this feature for the experiments reported in this chapter.

[10] This feature was described in much more detail in Section 7.2.2.5. There were no changes made to the definition of this feature for the experiments reported in this chapter.

IF	*the second phrase is an alias of the first phrase*	(ALIAS = *YES*)
AND	*neither or both phrases refer to a joint venture*	(XOR-JV-CHILD = *NO*)
THEN	class = **YES**	(the phrases are coreferent)

Figure 8.1 A rule with domain-independent and domain-specific features

8.4 Ablation Experiments

Ablation experiments can be performed by training and testing decision trees with some of the features disabled or ignored by the decision tree construction, pruning and classification algorithms. The differences in performance when different sets of features are disabled can provide some evidence for the relative importance of each set – if disabling one set of features results in dramatically worse performance than the results observed when another set of features is disabled, we can conclude that the former set of features is more important than the latter set.

It is possible to conduct ablation experiments on manually engineered systems, however manually modifying an existing rulebase can be much more difficult than retraining a decision tree. This issue is addressed in more detail in Section 8.4.1.

The preceding sections define a partition for the features used by RESOLVE in the MUC-5 EJV domain: a set of domain-independent features and a set of domain-specific features. The other sections below describe a series of ablation experiments performed on each of these sets of features.

8.4.1 Machine Learning vs. Manual Engineering

Manually constructing two systems in order to assess the relative importance of domain-independent knowledge would be quite difficult. It is quite likely that within a set of rules for coreference resolution, one would have rules that mix domain-independent knowledge and domain-specific knowledge. An example of one such rule, based on the decision tree shown in Figure 7.4, is shown in Figure 8.1.

Simply removing the domain-specific features from a rule that contains domain-independent features may result in individual rules that are no longer adequately constrained. Such rules would need to be rewritten or removed. The set of new rules may then need to be rearranged. Thus, an experiment to assess the relative importance of domain-independent and domain-specific knowledge would entail the manual construction of two different two rule-bases, requiring up to twice the effort as constructing a single rule-base.

Using a machine learning algorithm to combine and arrange the features, it is possible to assess the relative importance of domain-independent and domain-specific knowledge with little effort beyond what is required to define the features. Decision trees can be trained and tested with access to all of the features, and then trained and tested with access to only the domain-independent features; no manual intervention is necessary, beyond specifying the category (domain-independent or domain-specific) for each feature.

8.4.2 Experimental Methodology

A series of 10-fold cross-validation experiments was run in which different sets of features were disabled during training and testing of the decision trees used by RESOLVE. The cross-validation method will be described below in Section 8.4.2.1.

Three different strategies for selecting the features that would be disabled in any given experiment were employed; these will be described in Section 8.4.2.2.

8.4.2.1 10-Fold Cross-Validation

For each experiment, a set of eight features was disabled, and a 10-fold cross-validation test was performed on the instances generated from each of the 50 annotated MUC-5 EJV texts. The 50 sets of instances were randomly divided into ten partitions of five sets of instances each. For each of the ten cross-validation iterations, the instances from one partition (representing five texts) were selected as the test set and the instances in the other nine partitions (representing 45 texts) were used as the training set.

A C4.5 decision tree was constructed based on the instances in the training set, ignoring the disabled features. The decision tree was then simplified by the C4.5 pruning procedure. The instances in the test set were then classified by this simplified decision tree, and the recall and precision was computed for each of the 5 sets of instances in the test set.

A different partition was selected for the test set in each of the 10 iterations, so that by the end of an experiment, the instances from each of the 50 texts had been used as test instances once and training instances nine times. At the end of each experiment, 50 recall and precision scores were available.

8.4.2.2 Three Variations

Three different variations were used to select the features disabled under each series of experiments. These variations were motivated by the goal of assessing the relative importance of the domain-specific features to the coreference resolution task. The labels used for each variation described below indicate the number and types of features that are disabled under that variation.

1. *No Features*

   Under one strategy, no features were disabled, i.e., the decision trees were trained and tested with all features available to them. The only aspect that varied across each 10-fold cross-validation experiment under this variation was the partitioning of the instances into training and testing sets. It was assumed that the performance of the decision trees under this strategy would be better than under any other strategy.

2. *Any 8 Domain-Independent Features*

   Another strategy was to randomly select eight domain-independent features to be disabled during training and testing of the decision trees. For each 10-fold cross-validation experiment, a new set of domain-independent features was randomly selected, and the instances were randomly repartitioned into training and testing sets. All of the features used in these experiments were manually engineered with the goal of achieving good coreference resolution performance[11], therefore it was expected that disabling any group of eight domain-independent features would result in lower performance than could be achieved with all the features being used.

[11] Unlike the approach taken by WRAP-UP [Soderland and Lehnert, 1994], which treats all of the data coming from a sentence analyzer as potential features, and ends up with tens of thousands of automatically generated features.

Table 8.5 The impact of domain-specific knowledge on coreference resolution

Disabled Features	*Recall*		*Precision*	
	Mean	*Variance*	*Mean*	*Variance*
None	79.4	568.2	92.4	263.5
Any 8 Domain-Independent	76.5	600.4	91.5	294.2
All 8 Domain-Specific	59.9	965.0	83.7	710.2

3. *All 8 Domain-Specific Features*

   The final variation was to disable all eight domain-specific features during the training and testing of the decision trees. As with the first variation, in which no features were disabled, the only aspect that varied across each 10-fold cross-validation experiment under this variation was the partitioning of the instances into training and testing sets. As with the previous strategy of disabling a randomly selected set of domain-independent features, it was expected that performance under this variation would be lower than the performance of decision trees trained and tested with all features.

Eight domain-independent features were selected for the second variation since that was the total number of domain-specific features, and all of the domain-specific features were disabled under the third variation. The goal is to disable an equal number of domain-independent features as a control condition, to determine whether performance goes down merely due to the number of features available for training and testing, rather than due to the relative importance of the disabled features for coreference resolution.

For each variation, 100 cross-validation experiments were run. Since one recall score and precision score was collected for each of the 50 texts in each experiment, the 100 experiments yielded a total of 5000 data points for both recall and precision.

8.4.3 Results of the Experiment

As outlined in the previous section, 100 10-fold cross-validation experiments were run on the instances from 50 texts for each variation of disabled features. The mean and variance of recall and precision across all 5000 data points are shown in Table 8.5.

A sample C4.5 decision tree trained with *all* features and *all* instances is shown in Figure 8.2.[12] A decision tree trained with only the domain-independent features, i.e., without any domain-specific features, is shown in Figures 8.3 and 8.4.

8.4.4 Statistical Analysis of the Results

Scheffé tests (see Cohen [1995], Section 6A) were run on the data generated for these experiments in order to determine which of the changes in performance resulting

[12] The actual decision trees generated for the experiments under the ***no features*** variation may differ slightly from this decision tree, since each decision tree generated during these experiments was trained on instances from 45 texts rather than instances from the entire set of 50 texts. A similar caveat applies to the decision tree depicted in Figures 8.3 and 8.4.

```
X-IS-Y = Y: "+" (14.0/1.3)
X-IS-Y = N:
  ALIAS = Y:
    XOR-JV-CHILD = Y: "-" (10.0/3.5)
    XOR-JV-CHILD = N: "+" (42.0/1.4)
    XOR-JV-CHILD = U: "+" (68.0/7.3)
  ALIAS = N:
    BOTH-JV-CHILD = Y:
      JV-PARENT-2 = Y: "-" (3.0/1.1)
      JV-PARENT-2 = N: "+" (107.0/8.4)
      JV-PARENT-2 = U: "+" (0.0)
    BOTH-JV-CHILD = N:
      COMMON-LOC = N: "-" (42.0/1.4)
      COMMON-LOC = U: "-" (734.0/32.4)
      COMMON-LOC = Y:
        XOR-JV-CHILD = Y: "-" (14.0/2.5)
        XOR-JV-CHILD = N: "+" (12.0/2.5)
        XOR-JV-CHILD = U: "-" (0.0)
    BOTH-JV-CHILD = U:
      X-SAID-IT = Y: "+" (6.0/1.2)
      X-SAID-IT = N:
        PRONOUN-1 = Y:
          SAME-SENTENCE = Y: "-" (2.0/1.0)
          SAME-SENTENCE = N: "+" (14.0/7.8)
        PRONOUN-1 = N:
          SAME-NAME = Y: "-" (0.0)
          SAME-NAME = N: "-" (303.0/3.9)
          SAME-NAME = U:
            COMMON-NOUN = Y:
              NAME-2 = N: "+" (6.0/2.3)
              NAME-2 = Y:
                LOC-2 = Y: "-" (2.0/1.0)
                LOC-2 = N: "+" (3.0/2.1)
            COMMON-NOUN = N:
              COMMON-NM = Y:
                DEF-ART-1 = Y: "+" (2.0/1.0)
                DEF-ART-1 = N:
                  JV-PARENT-2 = Y: "-" (3.0/1.1)
                  JV-PARENT-2 = N: "-" (0.0)
                  JV-PARENT-2 = U: "+" (4.0/2.2)
              COMMON-NM = N:
                PRONOUN-2 = N: "-" (305.0/53.1)
                PRONOUN-2 = Y:
                  JV-PARENT-1 = Y: "-" (7.0/1.3)
                  JV-PARENT-1 = N: "+" (8.0/2.4)
                  JV-PARENT-1 = U: "-" (7.0/2.4)
```

Figure 8.2 A pruned C4.5 decision tree based on *all* features

```
ALIAS = Y:
  SAME-SENTENCE = N: "+" (114.0/12.8)
  SAME-SENTENCE = Y:
    SAME-STRING = Y: "+" (3.0/1.1)
    SAME-STRING = N: "-" (3.0/1.1)
ALIAS = N:
  X-IS-Y = Y: "+" (14.0/1.3)
  X-IS-Y = N:
    COMMON-NP = Y:
      DEF-ART-1 = Y: "+" (10.0/1.3)
      DEF-ART-1 = N: "-" (3.0/1.1)
    COMMON-NP = N:
      COMMON-LOC = N: "-" (94.0/1.4)
      COMMON-LOC = Y:
        SAME-SENTENCE = Y: "-" (8.0/1.3)
        SAME-SENTENCE = N:
          BOTH-SUBJECT = Y: "+" (3.0/1.1)
          BOTH-SUBJECT = N:
            GOVERNMENT-2 = Y: "-" (0.0)
            GOVERNMENT-2 = U: "+" (7.0/2.4)
            GOVERNMENT-2 = N:
              GOVERNMENT-1 = Y: "-" (0.0)
              GOVERNMENT-1 = U: "-" (2.0/1.0)
              GOVERNMENT-1 = N:
                INDEF-ART-1 = Y: "+" (9.0/4.5)
                INDEF-ART-1 = N: "-" (16.0/6.9)
      COMMON-LOC = U:
        PRONOUN-2 = Y:
          SAME-CONSTITUENT = Y:
            SAME-SENTENCE = Y: "+" (10.0/2.4)
            SAME-SENTENCE = N:
              PREVIOUS-SENTENCE = N: "-" (5.0/2.3)
              PREVIOUS-SENTENCE = Y:
                LOC-1 = Y: "+" (2.0/1.0)
                LOC-1 = N:
                  NAME-1 = Y: "-" (2.0/1.0)
                  NAME-1 = N: "+" (2.0/1.0)
          SAME-CONSTITUENT = N:
            INDEF-ART-1 = N: "-" (9.0/2.4)
            INDEF-ART-1 = Y:
              PREVIOUS-SENTENCE = Y: "-" (3.0/2.1)
              PREVIOUS-SENTENCE = N: "+" (4.0/2.2)
                {continued ...}
```

Figure 8.3 A pruned C4.5 decision tree based on *domain-independent* features

```
ALIAS = N:
  X-IS-Y = N:
    COMMON-NP = N:
      COMMON-LOC = U:
        PRONOUN-2 = N:
          SAME-SENTENCE = Y: "-" (365.0/8.5)
          SAME-SENTENCE = N:
            BOTH-GOVERNMENT = Y: "+" (2.0/1.0)
            BOTH-GOVERNMENT = U: "-" (498.0/60.6)
            BOTH-GOVERNMENT = N:
              LOC-1 = Y: "-" (159.0/26.8)
              LOC-1 = N:
                DEF-ART-2 = N: "-" (178.0/21.5)
                DEF-ART-2 = Y:
                  INDEF-ART-1 = Y:
                    LOC-2 = Y: "-" (10.0/4.6)
                    LOC-2 = N: "+" (45.0/13.6)
                  INDEF-ART-1 = N:
                    DEF-ART-1 = N: "-" (103.0/22.4)
                    DEF-ART-1 = Y:
                      BOTH-SUBJECT = Y: "+" (10.0/3.5)
                      BOTH-SUBJECT = N:
                        PREVIOUS-SENTENCE = Y: "-" (15.0/4.7)
                        PREVIOUS-SENTENCE = N:
                          LOC-2 = Y: "-" (3.0/1.1)
                          LOC-2 = N: "+" (7.0/3.4)
```

Figure 8.4 Continuation of decision tree in Figure 8.3

Table 8.6 Analysis of variance for recall scores

Source	df	Sum of Squares	Mean Square	F	p value
Between	2	1099905.7	549952.9	780.5	$p \leq .0001$
Within	14997	10567357	704.6		
Total	14999				

Table 8.7 Analysis of variance for precision scores

Source	df	Sum of Squares	Mean Square	F	p value
Between	2	227971.2	113985.6	269.7	$p \leq .0001$
Within	14997	6338306.5	422.6		
Total	14999				

from the different variations were statistically significant. The analysis of variance for the recall scores is shown in Table 8.6; the analysis of variance for the precision scores is shown in Table 8.7.

Based on these analyses of variance, we can contrast the recall and precision scores among each of the three pairs of variations using the formula:

$$F = \frac{C^2}{\sigma_C^2 (j-1)} = \frac{C^2}{MS_{within} \sum_i \frac{w_i^2}{n_i} (j-1)}$$

where

- C is the difference between the means between a pair of groups (each set of 5000 data points collected under a single variation represents a distinct group).
- MS_{within} is the Mean Square within (e.g., in Tables 8.6 and 8.7, this number can be found under the column headed by "Mean Square" and the row headed by "Within").
- w_i is a weight term, set to 1 in these comparisons.
- n_i is the number of data points for each group (variation) being compared.
- j is the number of groups (variations) being compared.

The three variations described in Section 8.4.2.2 will be referred to in the subscripts of F as *none*, *dom-ind* and *dom-spec*, respectively, denoting the set of features that were disabled in each set of experiments. For recall, the formula yields the following F values:

$$F_{none,\,dom\text{-}ind} = 26.9$$

$$F_{none,\,dom\text{-}spec} = 1218.1$$

$$F_{dom\text{-}ind,\,dom\text{-}spec} = 882.8$$

For precision, the following $F_{i,j}$ values were computed:

$$F_{none,\,dom\text{-}ind} = 4.8$$

$$F_{none,\,dom\text{-}spec} = 447.9$$

$$F_{dom\text{-}ind,\,dom\text{-}spec} = 360.0$$

Each of these values of F is statistically significant, given 2 degrees of freedom in one dimension and infinite degrees of freedom in the second dimension. From this analysis, we can conclude that recall and precision performance is significantly degraded whenever we disable eight of RESOLVE's features – whether these eight come from the set of domain-independent features or the set of domain-specific features. Furthermore, we can conclude that both recall and precision goes down significantly more when the eight domain-specific features are disabled than when any eight domain-independent features are disabled.

8.4.5 Discussion

The results shown in Table 8.5, and the statistical analyses presented in the previous section, demonstrate that, on average, there is a significant drop in RESOLVE's performance – in both recall and precision – whenever eight features are disabled. This result is not surprising, since each of the features used for these experiments was manually engineered with the goal of improving coreference resolution performance.[13]

Although performance tends to degrade whenever eight features are disabled, the degradation is much worse, on average, when the eight domain-specific features are disabled than when eight randomly selected domain-independent features are disabled. From this result, we can conclude that domain-specific features are very important to coreference resolution.

8.5 Why RESOLVE Fails to Achieve 100% Recall and 100% Precision

RESOLVE achieves higher recall when it is given access to all of its features than it does when its access is restricted to subsets of these features, e.g., either the domain-specific features (as shown in this chapter) or the small set of features drawn from the

[13]Contrast this with machine learning approaches to other NLP problems, e.g., part-of-speech tagging, wherein the features are based on words and part-of-speech tags, with little need for manual engineering of higher-level features.

coreference resolution rules of the UMass/Hughes MUC-5 system. The differences in precision are smaller than the differences in recall for the different sets of features, but even with the full set of features, RESOLVE is unable to achieve 100% precision.

It is worth noting that we do not know exactly how well people perform at coreference resolution. We might like to believe that reasonably intelligent human readers would achieve 100% recall and 100% precision on this task. We might also like to believe that people can achieve 100% recall and 100% precision on the information extraction task; however, a study conducted during the MUC-5 evaluation showed that professional analysts only achieve 80% recall and 80% precision on that task [Will, 1993].[14] Although the best performance of RESOLVE may not be perfect, it may be closer to human performance than might be expected.

We do not know how well humans perform on the coreference resolution task, however, the performance of RESOLVE has been measured, and we can examine where the system fails. Due to the symmetric and transitive properties of identity coreference, it is possible to misclassify some positive instances (false negatives) and still achieve 100% recall. For example, if a phrase is coreferent with three earlier phrases in a text, it is sufficient to find a coreferent link between the new phrase and just one of those earlier phrases, and rely upon a transitive closure operation to join all four phrases; two of the coreference links can be missed (false negatives) as long as the third is found (correctly classified as positive). Of the 330 positive instances in the MUC-5 EJV dataset, a minimum of 218 must be correctly classified in order to achieve 100% recall.

Although transitive closure can compensate for some false negative classifications, a false positive classification *always* affects precision, since it results in the joining of two closures that should remain separate (see Section 6.3 for examples of how false positives affect precision).

The ensuing discussion in this section and the following section focuses only on errors that affect recall and precision, rather than all misclassification errors. Therefore, the discussion centers on false positives and *missed links*, where the latter category includes those phrases which are coreferent with one or more earlier phrases, but for which no positive classifications were returned (i.e., phrases for which no coreference link was found).

The first section below will provide some high-level explanations for why RESOLVE fails to achieve 100% recall and 100% precision. The second section will provide more details on exactly where the system failed and why.

8.5.1 A General Discussion of Errors Made by RESOLVE

Even when all of the features are available for training, RESOLVE is still not able to achieve 100% recall or 100% precision. Table 8.8 shows that a total of 13 false positive errors were responsible for the system's imperfect precision, and a total of 37 missed links were responsible for the system's imperfect recall.

The data used for these experiments was relatively error-free, i.e., the annotation of phrases and the information about the phrases that was used to generate instances were accurate. However, some errors were introduced by the definition of some of the features. For example, the definition of the ALIAS feature introduced some *ambiguity* in the instances: the matching function used by this feature sometimes found "good" matches between a new phrase and more than one previous phrase (each of which referred to different organizations).

[14] Where the information extracted by one analyst was designated as the key template, and the information extracted by other analysts was designated as response templates.

Table 8.8 Breakdown of Errors

	Error Type	
Source of Error	*False Positives*	*Missed Links*
Feature Ambiguity	10	3
Incomplete Semantic Knowledge	2	12
Unused Features	1	15
Others	0	7
Totals	13	37

Another group of errors resulted from the unavailability of more detailed semantic knowledge. The information collected about the phrases used in these experiments was based on a rather small and shallow semantic hierarchy – one that distinguished organizations from people but did not distinguish between different types of companies (e.g., distinguishing financial institutions from manufacturing companies).

C4.5 did not use all of the features it had available in constructing its decision trees; the set of features actually used in the decision trees grew even smaller after pruning. Some of the features, or combinations of features, that were needed to classify pairs of phrases as coreferent were not present in the decision trees. This is due in part to the fact that some instances that could have been classified based on these *unused* features were able to be classified based on other combinations of features (which may or may not appear as intuitive to a knowledge engineer). Another explanation for these unused (but presumably important) features is that the pruning procedure that simplifies the decision tree is intended to discard combinations of features that are not likely to cover many [unseen] cases; the fact that the pruned decision trees correctly classified *most* of the positive instances is evidence that few truly important features were discarded.

There were other errors that do not fall into any of these more general categories; these will be described in greater detail in Section 8.5.2.4 below.

8.5.2 A Detailed Analysis of Errors Made by RESOLVE

A set of general categories of errors that were made by RESOLVE was presented in the previous section. This section will elaborate on these errors and provide specific examples that illustrate each error category.

8.5.2.1 Feature Ambiguity

Some joint venture companies inherit portions of the names of one or more of their parent companies. This can make the computation of the ALIAS feature quite difficult. Four out of the thirteen false positive errors made by RESOLVE were the result of ambiguity in aliases of joint ventures or their parent companies. Examples include determining whether

- `SUMITOMO` is an alias of `SUMITOMO CORP.` or `SUMITOMO ELECTRICAL INDUSTRIES LTD.` (the latter);
- `DAIWA` is an alias of `THE DAIWA BANK` or `DAIWA LIPPO` (the former);

- `IEC` is an alias of `Avon IEC` (it is not); or
- `Wickes` is an alias of `Wickes Manufacturing Co.` or `Wickes Cos.` (the latter).

Other sources of ambiguity involve assumptions made in determining whether a phrase refers to a joint venture company (JV-CHILD) or a joint venture parent company (JV-PARENT), or whether a pair of phrases both refer to a joint venture company (BOTH-JV-CHILD). The specific sources of ambiguity include:

- One instance of a joint venture parent company that was itself the joint venture child company of yet another business tie-up.
- One instance of a company that was originally introduced as a joint venture parent company but was later described as the parent company of two subsidiary companies that were the actual joint venture parent companies. The JV-PARENT feature was defined to be *NO* if a company was the sole parent of one or more subsidiaries, so this resulted in one reference having a JV-PARENT value of *YES* and another having a value of *NO*.
- Two texts referenced more than one joint venture. This resulted in two false positive errors (one in each text) in which the BOTH-JV-CHILD feature was used to link references to two distinct joint venture companies.

There were two examples where the COMMON-NOUN feature had a value of *YES* due to a match on the pronoun `IT`. While two pronominal references are often coreferential, especially when they occur in adjacent sentences [Brennan *et al.*, 1987], pronouns probably should have been excluded in the definition of this feature (especially since there were explicit PRONOUN-i features to capture this kind of information).

All of the features based on string matching (Section 8.2.2) and noun phrase analysis (Section 8.2.5) might benefit from additional constraints based on domain-specific knowledge, e.g., a match on words that denote generic organization descriptors (Table 8.3), or corporate designator abbreviations (Table 8.4) may not be indicative of coreference – more than one company may have a name that ends with the designator `CO.` or `CORP.`

8.5.2.2 Incomplete Semantic Knowledge

The CMI annotation tool was used to collect the data used in the experiments reported in this chapter. The goal of using this tool was to eliminate errors from the input to RESOLVE, errors that would likely occur if a sentence analysis program were used to collect the data. By using CMI, the credit assignment is simplified – all errors are the result of coreference processing and not, for example, the result of incorrect part-of-speech assignments.

However, since RESOLVE was intended for use in conjunction with a sentence analyzer, e.g., the CIRCUS sentence analyzer, the information included with the CMI annotations was restricted to the type of information that *could*, in principle, be generated by a sentence analyzer. Some information that would be useful for coreference resolution, but was not available from CIRCUS, was not included in the information.

For example, CIRCUS was able to distinguish references to people from references to companies. When ported to the MUC-5 EJV domain, the sentence analyzer could even distinguish between references to joint venture children and references to the parents of joint ventures. However, the system was not capable of finer levels of distinction, e.g., distinguishing manufacturing companies from financial institutions.

In order to achieve finer levels of distinction, an extensive semantic hierarchy would be required.

For example, knowledge about different types of companies, e.g., manufacturing companies vs. financial institutions, would have been useful in resolving THE JAPANESE MAKER in the third sentence below:

> OSAKI ELECTRIC CO., A MANUFACTURER OF POWER DISTRIBUTION EQUIPMENT, SAID THURSDAY IT HAS SET UP A JOINT COMPANY IN INDONESIA TO PRODUCE INTEGRATING WATT-HOUR METERS.
>
> BASED IN DJAKARTA, THE NEW FIRM, CALLED METBELOSA, IS CAPITALIZED AT 2.5 MILLION DOLLARS, OF WHICH 44 PCT WAS PUT UP BY OSAKI, 30 PCT BY METRIKA OF INDONESIA, 11 PCT BY KANEMATSU-GOSHO, LTD., A JAPANESE TRADING HOUSE, AND THE REMAINDER BY OTHER INTERESTS.
>
> IT WILL OPERATE METRIKA'S IDLE PLANT IN THE INDONESIAN CAPITAL TO TURN OUT 200,000 WATT-HOUR METERS IN THE INITIAL YEAR WITH PARTS SUPPLIED BY THE JAPANESE MAKER.

In this text, RESOLVE failed to link THE JAPANESE MAKER (in the third sentence) with OSAKI ELECTRIC CO., A MANUFACTURER OF POWER DISTRIBUTION EQUIPMENT (in the first sentence). This link might have been found if the two phrases were both marked as referring to manufacturing companies. Another example can be seen in the text below, where the two references to the new beverage company, A JOINT VENTURE TO PRODUCE BEVERAGES and THE BEVERAGE COMPANY, were not linked by the system.

> LIEM SIOE LIONG, A BUSINESS TYCOON IN INDONESIA, AND YAKULT HONSHA CO. OF JAPAN PLAN TO ESTABLISH
> A JOINT VENTURE TO PRODUCE BEVERAGES WITH AN INVESTMENT OF 3 MILLION DOLLARS NEXT MONTH, THE JAKARTA POST SAID.
>
> THE NEW FIRM, PT YAKULT INDONESIA PERSADA, WILL BE 51 PCT OWNED BY LIEM AND 49 PCT BY YAKULT, AND WILL BE BASED IN JAKARTA, WHILE ITS PLANT WILL BE BUILT IN BOGOR, WEST JAVA, WITH A DESIGNED CAPACITY OF ABOUT NINE MILLION BOTTLES OF LACTIC ACID BACTERIUM BEVERAGES A MONTH.
>
> THE BEVERAGE COMPANY IS EXPECTED TO START PRODUCTION NEXT YEAR AND WILL SELL ITS PRODUCTS MAINLY ON THE DOMESTIC MARKET.

Other distinctions that would have been useful in the training corpus include identifying companies whose business had to do with:

- automobiles
  - TAIWAN'S LARGEST CAR DEALER
  - CHINESE AUTOMOBILE CO.
- metals
  - SUMITOMO SPECIAL METALS CO.,

- THE SUBSIDIARY OF SUMITOMO METAL INDUSTRIES LTD., A MAJOR JAPANESE STEELMAKER

- construction
  - AOKI CONSTRUCTION CO.
  - THE JAPANESE CONSTRUCTION FIRM

However, a much more extensive semantic hierarchy would need to be developed in order for RESOLVE to be able to handle novel references to companies in other lines of business.

Another source of incomplete knowledge was the restricted inferences regarding the location of entities. The MUC-5 EJV task definition specified that only explicitly mentioned location information about any entities could be included in the output template for a text. Among the implicit location information that was disallowed in this task were things like inferring that a company name that started with the designator P.T., e.g., P.T. SAPTA PANJI MANGGALA, was a reference to an Indonesian company, even though the initials "P.T." stand for the Indonesian words "Persoroan Terbatas", and are only used to designate Indonesian companies throughout the MUC-5 EJV corpus.

Another type of inference that was not permitted was to extract location information from some company names, e.g., it was not permissible to infer that PROTON AMERICA, INC. was a reference to a company based in the United States.[15]

Since these sorts of location inferences were disallowed under the MUC-5 EJV task definition, they were not made during the course of CMI annotations. Some of these inferences would have proven useful in resolving references in this domain, however.

Over one third (11 out of 37) of the missed links were the result of semantic information not being available for coreference resolution.

8.5.2.3 Unused Features or Feature Combinations

Some features occurred in a rather small subset of the training data, and were either absent from the decision trees or present in some subtree not visited during classification of an instance where that feature was crucial for coreference resolution.

Certain combinations of features would have been useful for classifying some phrases as coreferent. For example, the most likely way to resolve the two references to each of two joint venture parent companies in the sentences below is to note that each occurs in the same constituent buffer (subject or prepositional phrase) of their respective sentences, that is **The Italian apparel concern** might be linked with **Benetton Group S.p.A.** because they occur as the subjects of their respective sentences, and **the giant Japanese retailer** might be linked to **the Seibu/Saison Group** because both occur in prepositional phrases.

> Benetton Group S.p.A. is determined to consolidate its presence in the Japanese market by turning its licensing agreement with the Seibu/Saison Group into a joint-venture accord.
>
> The Italian apparel concern hopes to conclude negotiations with the giant Japanese retailer before the end of the year, a spokesman said.

[15] Of course, such inferences are not always correct anyhow, since CHINESE AUTOMOBILE CO. is a reference to a Taiwanese car dealer.

The features BOTH-SUBJECT or SAME-CONSTITUENT, when coupled with the PREVIOUS-SENTENCE, would have enabled correct classification of these two pairs of phrases.

There were several other examples where the combination of BOTH-SUBJECT and PREVIOUS-SENTENCE could have enabled the correct classification of a pair of phrases as coreferent. In some cases, the learning algorithm was able to capture other relationships between the phrases; in other cases, no coreference link could be established. The use of a greedy algorithm is not guaranteed to produce an "optimal", or in some cases even "good", combinations of features.

8.5.2.4 Other errors

Three missed links might have been found by the use of ordering information, i.e., whether a phrase was the first, second, and so on, that occurred in a conjunction. For example, in the two paragraphs below, the only way to link **Shell** with **Compagnie Francaise des Petroles** is that they each occur as the second conjunct in two conjunctions:

> **QATAR is close to finalising agreement with British Petroleum and Compagnie Francaise des Petroles on implementing a $4bn (2.8bn) project for exploiting the state's off-shore North gas field.**
>
> **BP and Shell were selected just over a year ago as prospective partners with a 7 1/2 per cent share each in any liquified natural gas (LNG) venture, from a number of interested companies including also Japan and Taiwan.**

Another possible explanation for the way that these sets of phrases can be correctly resolved has to do with a process of elimination. A human reader likely links **BP** and **British Petroleum**, and then the only remaining possible antecedent for **Shell** is **Compagnie Francaise de Petroles**. This process of elimination would be very difficult to model within the representation selected for RESOLVE, but may be possible under other representations.[16]

8.6 Why Domain-Specific Knowledge is Important

The average degradation in performance that resulted from disabling the domain-specific features has been discussed in Section 8.4.3. Having established *that* domain-specific features are important for coreference resolution, the purpose of this section is to provide some explanation as to *why* domain-specific features are useful.

Knowledge specific to the MUC-5 EJV domain was important in two fundamental ways: some coreferent phrases could be correctly classified only on the basis of domain-specific features, and the ambiguity of positive evidence for coreference provided by some domain-independent features was greatly reduced by the addition of domain-specific features. These two effects of domain-specific knowledge will be examined in more detail in the following sections.

[16] Different representations for the problem were discussed in Section 4.1.1.

```
X-IS-Y = N:
  ALIAS = N:
    BOTH-JV-CHILD = Y:
      JV-PARENT-2 = N: "+"

SUB-STRING = N:
  X-IS-Y = N:
    ALIAS = N:
      BOTH-JV-CHILD = Y:
        JV-PARENT-2 = N: "+"

ALIAS = N:
  BOTH-JV-CHILD = Y:
    JV-PARENT-2 = N: "+"

X-IS-Y = N:
  ALIAS = N:
    BOTH-JV-CHILD = Y: "+"
```

Figure 8.5 BOTH-JV-CHILD as a *Key* Feature

8.6.1 Coreference based on Domain-Specific Features Only

The BOTH-JV-CHILD feature was the *key* feature used by decision trees to correctly classify 62 positive instances, accounting for 28% of the 218 positive instances that must be correctly classified to achieve 100% recall. A *key* feature is one that has a positive (*YES*) value in a decision tree node, and all ancestors of the decision tree have features with negative (*NO*) values.[17] The BOTH-JV-CHILD was also the key feature used to incorrectly classify 3 negative instances (false positives).

During a 50-fold cross-validation experiment, four examples of decision trees were found in which BOTH-JV-CHILD was a key feature.[18] Figure 8.5 shows the relevant portions of each of these decision trees. In each case, the ancestors of the decision tree node that tests BOTH-JV-CHILD have negative values, and the positive classification is based primarily on the positive value associated with the BOTH-JV-CHILD node.

When the BOTH-JV-CHILD feature was disabled along with the other domain-specific features, some of the domain-independent features were able to compensate for some of the coreference links no longer found on the basis of identifying joint venture companies.[19] Of the 62 positive instances that had been correctly classified on the basis of the BOTH-JV-CHILD feature, 36 were correctly classified on the basis of three domain-independent features: 15 were found by the COMMON-NM feature

[17] Nearly all of the features defined for RESOLVE are defined such that a positive value is indicative of coreference; negative values do not usually imply non-coreference (though a series of negative values may do so).

[18] The variations are due to the fact each decision tree was trained on a slightly different set of instances drawn from 49 texts.

[19] This ability of domain-independent features to compensate for missing or poorly defined domain-specific features will be discussed in greater detail in Section 9.3.

(matching on noun modifiers such as NEW or JOINT, often used to describe joint venture companies); 9 were found by the COMMON-NOUN feature (matching the noun VENTURE); and 12 were found by the SUB-STRING feature (matching the phrase JOINT VENTURE).

Although 60% of the positive instances that had been correctly classified with the BOTH-JV-CHILD feature can also be correctly classified on the basis of domain-independent features, the remaining 40% of such instances could not be correctly classified by RESOLVE when the domain-specific features were disabled. In the following text fragment, the link between the phrases A JOINT VENTURE (in the first sentence) and ALCOM NIKKEI SPECIALTY COATINGS SDN. BHD. (in the second sentence) is missed, since no information is available about both phrases referring to joint venture companies; this link was correctly identified by RESOLVE when it had access to the BOTH-JV-CHILD feature.

> NIPPON LIGHT METAL CO. HAS LAUNCHED A JOINT VENTURE IN MALAYSIA TO PRODUCE AND SELL ALUMINUM PRECOATED FINS, WITH ALUMINIUM CO. OF MALAYSIA BHD. (ALCOM), A SUBSIDIARY OF ALCAN ALUMINUM LTD. OF CANADA, NIPPON LIGHT METAL ANNOUNCED WEDNESDAY.
>
> ALCOM NIKKEI SPECIALTY COATINGS SDN. BHD. WILL START PRODUCTION IN OCTOBER NEXT YEAR, COMPANY OFFICIALS SAID.
>
> ...
>
> THE JOINT VENTURE WILL BE LOCATED IN THE BUKIT RAJA INDUSTRIAL PARK NEAR KUALA LUMPUR, WHERE ALCOM HAS A ALUMINUM PLATE MANUFACTURING PLANT.
>
> IT IS CAPITALIZED AT 12 MILLION RINGGITS, EQUALLY PUT UP BY NIPPON LIGHT METAL AND *ALCOM*.

The coreference link between THE JOINT VENTURE (third sentence) and IT (fourth sentence) is another example of a link being missed by RESOLVE when its domain-specific features are disabled. The link between these phrases was correctly identified by the system when it had access to the domain-specific features (in particular, the BOTH-JV-CHILD feature).

Of the 33 links that were missed by RESOLVE when its domain-specific features were disabled (but that were correctly identified by the system when it had access to those features), 26 were due to the unavailability of BOTH-JV-CHILD, 2 were due to the unavailability of BOTH-JV-PARENT, 3 were due to spurious branches descending from decision tree nodes in which ALIAS = *YES* (see more discussion on this topic in the next section) and 2 were for other reasons.

One last thing to note in the text fragment above is that it provides an example of a domain-independent feature compensating for a missing domain-specific features – RESOLVE finds the link between the phrases A JOINT VENTURE (first sentence) and THE JOINT VENTURE (third sentence) based on the COMMON-NM feature.[20]

[20]The SUB-STRING and COMMON-NOUN features would also have positive values for this instance, but they did not appear in the path of the tree used for the positive classification in this case.

8.6.2 Coreference based on Domain-Independent and Domain-Specific Features

The text fragment from the previous section illustrates another general category of errors made by RESOLVE when its domain-specific features are disabled. There are cases where positive evidence of coreference provided by domain-independent features is constrained or restricted by negative evidence provided by domain-specific features (and vice versa).

In the MUC-5 EJV domain, a positive value for the ALIAS feature did not always entail a coreferent relationship between phrases. This is due to the tendency of some joint venture companies to take on parts of the names of one or more of their parent companies. This phenomenon is seen in the text fragment, wherein the name of the joint venture company `ALCOM NIKKEI SPECIALTY COATINGS SDN. BHD.` includes a shortened form or alias of the one of its parent companies, `ALUMINIUM CO. OF MALAYSIA BHD. (ALCOM)`.

One way of eliminating the alias ambiguity in this and many other cases is by checking to see whether the phrases to be classified refer to joint venture companies or not. If one phrase refers to a joint venture company (JV-CHILD-i) and a potential alias of that phrase refers to a joint venture parent (JV-PARENT-i), the phrases are unlikely to be coreferent. In the text fragment from the previous section, `ALCOM` (fourth sentence) is a potential alias of either `ALUMINIUM CO. OF MALAYSIA BHD. (ALCOM)` (first sentence) or `ALCOM NIKKEI SPECIALTY COATINGS SDN. BHD.` (second sentence). If the system knows that `ALCOM` is a reference to a joint venture parent (an inference that can be drawn from its being one of the organizations capitalizing the joint venture), and that `ALCOM NIKKEI SPECIALTY COATINGS SDN. BHD.` is a reference to a joint venture child, then it can conclude that these phrases are not coreferent.

RESOLVE learns to constrain the use of the ALIAS feature via the XOR-JV-CHILD feature[21] to reflect this intuition. The decision tree paths shown in Figure 8.6 are taken from the same decision trees as those depicted in Figure 8.5. In each case, a positive value for the ALIAS feature does not result in a positive classification for instances in which one phrase is a reference to a joint venture company and the other phrase is not.

When RESOLVE is given access to its domain-specific features, it is able to correctly link `ALCOM` with `ALUMINIUM CO. OF MALAYSIA BHD. (ALCOM)`, since this instance has a positive value for the ALIAS feature and a negative value for the XOR-JV-CHILD feature (both phrases refer to joint venture parents). When the domain-specific features are disabled, the system is not able to learn useful restrictions on the ALIAS feature – sometimes it has no restrictions, i.e., a positive value for ALIAS results in a positive classification, in other decision trees, spurious distinctions are made in paths descending from a node in which ALIAS has a positive value. In the decision tree trained for classifying the text containing the four sample sentences in the previous section, a positive value for the ALIAS feature resulted in link being posited between `ALCOM` and `ALCOM NIKKEI SPECIALTY COATINGS SDN. BHD.`, since the node descending from ALIAS = *YES* was a leaf.

All of the false positive classifications that were made by RESOLVE *without* domain-specific features, but which were correctly classified (as negative instances) by RESOLVE *with* domain-specific features were the result of the ALIAS feature, or similar features such as SUB-STRING or COMMON-NM, not being appropriately constrained by domain-specific knowledge (usually the XOR-JV-CHILD feature).

[21] XOR-JV-CHILD has a positive value only when one phrase has a positive value for JV-CHILD-i and a negative value for JV-CHILD-j; $i, j \in \{1, 2\}, i \neq j$.

```
X-IS-Y = N:
  ALIAS = Y:
    XOR-JV-CHILD = Y: "-"
    XOR-JV-CHILD = N: "+"
    XOR-JV-CHILD = U: "+"

SUB-STRING = N:
  X-IS-Y = N:
    ALIAS = Y:
      XOR-JV-CHILD = Y: "-"
      XOR-JV-CHILD = N: "+"
      XOR-JV-CHILD = U: "+"

ALIAS = Y:
  XOR-JV-CHILD = Y: "-"
  XOR-JV-CHILD = N: "+"
  XOR-JV-CHILD = U: "+"

X-IS-Y = N:
  ALIAS = Y:
    XOR-JV-CHILD = Y: "-"
    XOR-JV-CHILD = N: "+"
    XOR-JV-CHILD = U: "+"
```

Figure 8.6 ALIAS and XOR-JV-CHILD as *Key* Features

CHAPTER 9

RESOLVE IN AN INFORMATION EXTRACTION SYSTEM

The preceding chapters have all focused on the use of RESOLVE as a stand-alone system. The data used for training and testing the system was collected via the CMI interface from texts in the MUC-5 EJV corpus. This configuration of the system was useful for conducting research into certain aspects of applying machine learning techniques to coreference resolution. However, a stand-alone configuration did not permit evaluating whether RESOLVE could function as a coreference resolution module within a larger information extraction system.

This chapter presents a case study in which RESOLVE was integrated with an information extraction system developed for the Sixth Message Understanding Conference [MUC-6, 1995]. Some of the constraints imposed for the work reported in previous chapters were relaxed when RESOLVE was employed for coreference resolution in MUC-6; these changes are described in the first section below.

Other changes required to use RESOLVE in MUC-6 included the annotation of a new set of texts in order to collect new training instances for the system and the development of new features to account for the relaxed constraints. These developments will be described in Section 9.2.

An interesting observation was made in the decision tree trained for the MUC-6 domain: domain-independent features were combined in ways that capture patterns that are specific to the MUC-6 domain (news articles about corporate management changes). Such patterns may be unintuitive, and perhaps even a bit risky, but they may prove useful to coreference resolution. One such pattern is examined in detail in the third section below.

The chapter concludes with a brief description of how RESOLVE was used other portions of the MUC-6 evaluation.

9.1 The MUC-6 Coreference Task

The coreference task definition generated a great deal of discussion before, during and after the official MUC-6 evaluation. Some program committee members wanted the coreference task to be tightly linked to the other tasks, e.g., candidate phrases would be restricted to those entities that had been identified during named entity recognition processing. Other members wanted to decouple coreference from the other tasks, to construe it as a broader task, evaluating coreference processing in general, not just for the references that were of potential value to later stages of processing in an information extraction system.

In the end, the proponents of a broadly construed coreference task won out, and most nouns, noun phrases and pronouns were candidates for coreference resolution.

9.1.1 New Challenges for RESOLVE

There were several significant differences between the type of task that RESOLVE was designed to perform and the type of task defined by the MUC-6 program committee. This is because RESOLVE was intended to work with an information extraction system – a system that produces candidates that are relevant to a specific information task – whereas the MUC-6 coreference candidates were completely unrelated to any specific information extraction task.

- *Relevant Entities*

  The phrases used for training and testing RESOLVE in the MUC-5 EJV domain were all references to entities relevant to the MUC-5 information extraction task; some references to companies and other types of entities were ignored, since they presumably would not be extracted by an information extraction system in the first place (see Section 5.2.1. This restriction had to be relaxed for MUC-6: all phrases potentially referring to *any* person or organization were candidates for coreference resolution. Note that this still represents a restriction with respect to the MUC-6 coreference task, since phrases referring to other types of entities were filtered out.

- *Relevant References*

  Not only were phrases referring to irrelevant entities ignored for the MUC-5 EJV domain, but phrases referring to relevant entities that did not contribute any [new] information about those entities were also ignored (see Section 5.2.2). This restriction was eliminated for MUC-6 – *any* reference to a person or organization was included in the training set for RESOLVE.

- *Full Noun Phrases*

  The training and testing data from the EJV domain was composed of full noun phrases – if there were any modifying nouns in the middle of a larger noun phrase that referred to an entity that was distinct from the referent of the larger noun phrase, the modifying nouns were ignored. This restriction was based on the limitations of the noun phrase analysis that was available in the UMass/Hughes MUC-5 system. Such analysis was able to extend simple noun phrases to include appositives and prepositional phrases which could be attached to the leading phrase; however, it did not attempt to analyze simple noun phrases in order to extract sub-phrases from the modifiers.

- *Appositives and Conjunctions*

  Appositive expressions and conjunctions were considered single units for the application of RESOLVE to the MUC-5 EJV domain. This was because these constructs were handled by earlier processing stages in the UMass/Hughes MUC-5 system. However, the MUC-6 Coreference Task Definition specified that the distinct components of an appositive expressions and conjunctions were to be separately annotated. For example, in the training text 930420015, the phrase

  ```
  Cecil R. Hash, chairman and chief executive officer
  ```

  would result in three different candidates

  ```
  Cecil R. Hash
  chairman
  chief executive officer
  ```

all of which would be coreferent.[1]

Since appositives and conjunctions would be handled by BADGER [Fisher *et al.*, 1995], adhering to this stipulation of the guidelines would require a post-processing step to split up appositives and conjunctions prior to the generation of a system response file.

9.1.2 MUC-6 Coreference Task Training Material

The MUC-6 training materials included a total of 225 texts that had been annotated with COREF SGML tags.[2] Unfortunately, there were many potential problems with using these texts as training for RESOLVE.

9.1.2.1 Different Versions of Coreference Task Definition

The MUC-6 Coreference Task Definition was being incrementally refined throughout the period preceding the release of the official training materials. Seven different versions of these guidelines were represented among the 225 annotated texts – and the last two versions of the guidelines were not represented by any annotated texts. Each new version of the guidelines added some new specifications for candidates for coreference resolution, deleted some old specifications, and modified others. This lack of consistency across the corpus would have created problems for most machine learning algorithms.

9.1.2.2 Inter-annotator Agreement

A number of MUC-6 participants collaborated on the annotation of texts for the Coreference Task. Each of these annotators brought his or her own background, motivation and interpretation of some version of the guidelines to the annotation task. This diversity among the annotators would have resulted in further inconsistencies among the different annotated texts.

9.1.2.3 No Additional Information about the Phrases

The only information contained in the annotated texts was bracketing SGML tags that indicated the scope of each candidate for coreference resolution (noun, noun phrase or pronoun), and a pointer to another SGML-delimited reference with which the candidate corefers. Some name information was potentially available, but only for the 30 texts that had also been annotated for the Named Entity Task, and those annotations were in separate files and would have had to be merged somehow with the Coreference Task annotations.

---

[1] Though the coreference link between the two conjuncts, **`chairman`** and **`chief executive officer`**, was considered ***optional***, i.e., systems were not penalized for missing conjunctive coreference links.

[2] SGML, or Standard Generalized Markup Language, is a notational scheme for bracketing segments of a document. See Appendix C for examples of the SGML tags used for MUC-6.

9.1.2.4 A Broad Definition of Coreference Candidates

As was mentioned earlier, the MUC-6 Coreference Task Definition provided very broad guidelines for which nouns, noun phrases and pronouns were to be considered candidates for coreference resolution. Since RESOLVE was only going to be trained on references to people and organizations – and was expecting that references to other entities would be filtered out during testing – the annotations for references to other types of entities would either have to be deleted or ignored.

9.1.2.5 Different Domain

Although the MUC-6 Coreference Task was designed to be domain-independent, RESOLVE was intended to be used with the rest of the UMass MUC-6 system, since coreference resolution was an important component of two of the other tasks in the evaluation. The training material for the Coreference Task was drawn from the domain of labor negotiations. Even if this training material were used to train RESOLVE for the Coreference Task, new training material from the domain used in the final evaluation (corporate management changes) would have had to been prepared for RESOLVE to work with other information extraction components on the other tasks.

9.2 RESOLVE in MUC-6

Despite the differences in orientation between the MUC-6 coreference task and the sort of task that RESOLVE was intended to perform, RESOLVE was applied to the MUC-6 coreference task. Unfortunately, due to the severe time limitations imposed by the task, very little time was available either for collecting training material or for developing new, domain-specific features for the corporate management changes domain.

One day was spent retargeting CMI for the new domain and then annotating training texts using the interface. Only 25 texts were annotated – some of these were processed twice, as new distinctions were made in the declarative knowledge that specified what was to be annotated and what types of information was to be collected about each reference.

Another day was spent developing some new features to capture some of the information relevant for resolving references in this new domain. One set of features was dedicated to the resolution of pronominal references; there were very few relevant pronominal references in the MUC-5 EJV domain, and most of those were resolved either using the X-SAID-IT feature or the contextual portions of the JV-CHILD-i features. However, since there was no relevancy filter used for the MUC-6 Coreference Task, nearly all pronouns were candidates for RESOLVE.

9.2.1 Using Constraints from the Named Entity Task

The broadly defined Coreference Task for MUC-6 seemed beyond the capabilities of RESOLVE, given the fact that it would be used in conjunction with a sentence analyzer (BADGER) that did not have the general world knowledge that would be required for resolving some of the references considered candidates for the task. Approximately 50% of the candidate phrases were references to organizations or persons, and another 20% were references to locations, times, dates, percentages and monetary amounts. BADGER would have to identify such phrases in order to handle other tasks within the MUC-6 evaluation.

We decided to apply RESOLVE to the coreference task, but to attempt to resolve only those phrases for which we had information – references to organizations and persons. Thus, we expected a maximum recall of 50% on the task. However, very little time was available to develop new domain-specific features for the MUC-6 domain. Since RESOLVE was able to achieve 57% recall in the MUC-5 domain when it was restricted to only using domain-independent features (see Table 8.5), we expected to achieve no more than the 30% recall when RESOLVE was run without domain-specific features on the MUC-6 domain. Another factor that further diminished our expectations with regard to RESOLVE's performance on this task was that some post-processing would need to be done after RESOLVE made its classifications, since appositives and conjunctions would need to be split prior to inserting SGML tags in the system response files.

If RESOLVE were given perfect data by BADGER, the highest scores we would expect to achieve would thus be 50% recall and 70% precision. Of course, BADGER is not a perfect sentence analyzer, and given the added complication of the special handling of appositives and conjunctions, we never expected to achieve more than 75% of those scores, i.e., a maximum recall of 38% and a maximum precision of 53%.

Since we anticipated our recall score to be so low, we decided to use the unpruned version of the decision trees generated by RESOLVE. These trees always had higher recall than any of the pruning variations during all of our tests, and we decided to attempt to maximize our recall at the potential expense of lower precision.

9.2.2 Training RESOLVE for MUC-6

Some of the issues involved in using the training material provided for MUC-6 were discussed in Section 9.1.2. The time and effort that would have been required to modify and use that material to train RESOLVE were daunting, and the lack of consistency and difference in domains may have rendered the resulting training instances less than useful.

However, the CMI tool was already available for annotating texts and automatically converting the annotations into instances that could be used for training and testing RESOLVE. The advantages of using this approach were considerable:

1. The most recent version of the guidelines could be used to mark candidate phrases, eliminating one source of inconsistency.

2. A single annotator could do all the work, eliminating another source of inconsistency.

3. Additional information about the phrases could be collected via features already included in the interface.

4. Candidate phrases could easily be restricted to those that referred to people and organizations.

5. The training material could be drawn from the same domain that was being used for the final evaluation for the Coreference Task, as well as the other tasks (including the domain-specific Scenario Template task).

The only potential drawback was the time required to use the interface to make the annotations. However, since a considerable amount of time would have been required to modify the existing training material for use with RESOLVE, it did not seem that there would be any net loss in time spent on training, and using CMI may have even provided a net gain in time available for other pressing tasks.

A total of 8 hours was spent on preparing training material with CMI. Retargeting the interface was a simple matter of declaring which types of entities were being referenced, and what types of information to collect about each reference. The remainder of the time (approximately 6 hours) was spent annotating a set of texts at the rate of approximately one every 15 minutes.

A set of 100 training texts was supplied for the MUC-6 Scenario Template task, half of which were relevant to the task. Of these 50 relevant texts, 37 had lengths of less than 2500 bytes – longer texts are more difficult (and time-consuming) to annotate. A set of 25 texts was selected at random from this set of 37 short, relevant texts. These 25 texts were annotated with CMI, generating a total of 227 references to 70 organizations and 159 references to 68 people.

The 386 references to organizations and persons were paired (within each text) to generate a total of 3482 instances, of which 834 (24%) were positive instances and the remaining 2648 (76%) were negative. These 3482 instances were used to train RESOLVE for the MUC-6 Coreference Final Evaluation.

9.2.3 Features Used for MUC-6

Most of the features defined for the MUC-5 EJV domain were also used for the MUC-6 domain. Some modifications to the feature extraction code had to be made in order to allow the features to be computed from either CMI-generated annotated references or BADGER system output.

The rest of this section describes modifications made to some EJV features to make them better applicable to the new domain and task definition in MUC-6 and some new features created to try to capture knowledge that is required for the new, broader task definition.

9.2.3.1 PARENT-i

Does phrase i refer to a parent organization?

Possible values: *YES, NO, UNKNOWN*

The PARENT feature was given the value of *YES* whenever a phrase contained the word `PARENT`; it was given the value of *NO* if the phrase referred to a child organization (see Section 9.2.3.2); otherwise it was given the value of *UNKNOWN*.

9.2.3.2 CHILD-i

Does phrase i refer to a child organization?

Possible values: *YES, NO, UNKNOWN*

The CHILD feature was given the value of *YES* whenever one of the words {`UNIT, UNITS, SUBSIDIARY, SUBSIDIARIES`} was detected in a phrase, or when there was a possessive form of an organization name in a phrase ; it was given the value of *NO* if the phrase referred to a parent organization (see Section 9.2.3.1); otherwise, it was given the value of *UNKNOWN*.

Table 9.1 Distribution of Feature Values for MUC-6

	Attribute Values					
	Positive Instances			Negative Instances		
Attribute Name	YES	NO	UNK.	YES	NO	UNK.
DEF-ART-1	0	834	0	0	2648	0
INDEF-ART-1	0	834	0	0	2648	0
PRONOUN-1	185	649	0	367	2281	0
LOC-1	34	800	0	174	2474	0
NAME-1	427	407	0	1441	1207	0
GOVERNMENT-1	1	357	476	17	978	1653
PARENT-1	14	21	799	83	73	2492
CHILD-1	21	14	799	73	83	2492
DEF-ART-2	0	834	0	0	2648	0
INDEF-ART-2	0	834	0	0	2648	0
PRONOUN-2	284	550	0	328	2320	0
LOC-2	18	816	0	119	2529	0
NAME-2	250	584	0	1346	1302	0
GOVERNMENT-2	0	232	602	18	597	2033
PARENT-2	10	7	817	75	78	2495
CHILD-2	7	10	817	78	75	2495
SAME-SENTENCE	124	710	0	291	2357	0
PREVIOUS-SENTENCE	203	631	0	580	2068	0
SAME-CONSTITUENT	384	450	0	905	1743	0
BOTH-SUBJECT	309	525	0	559	2089	0
SAME-STRING	110	724	0	4	2644	0
SUB-STRING	124	710	0	23	2625	0
COMMON-NOUN	158	676	0	35	2613	0
COMMON-NM	18	816	0	60	2588	0
COMMON-NM/NOUN	179	655	0	166	2482	0
COMMON-NP	165	669	0	44	2604	0
COMMON-LOC	0	0	834	3	8	2637
BOTH-GOVERNMENT	0	160	674	1	206	2441
ALIAS	121	713	0	1	2647	0
X-SAID-IT	11	823	0	0	2648	0
X-IS-Y	7	827	0	1	2647	0
SAME-TYPE	834	0	0	1117	1531	0
SAME-NUMBER	436	1	397	801	239	1608
SAME-GENDER	31	0	803	20	0	2628
MOST-RECENT-COMPATIBLE (MRC)	90	744	0	43	2605	0
MRC-NAMED	69	765	0	53	2595	0
MRC-SUBJECT	61	773	0	6	2642	0
MRC-NAMED-SUBJECT	95	739	0	25	2623	0
PERSON-IS-ROLE	12	822	0	0	2648	0

Table 9.2 Pronouns identified for MUC-6

Nominal	IT, THEY, HE, HIM, SHE, HER, I, WE, YOU
Possessive	ITS, THEIR, HIS, HERS, MY, OUR, YOUR

9.2.3.3 PRONOUN-i

Is phrase i a pronominal reference?

Possible values: *YES, NO*

For the EJV domain, the only relevant pronominal references were phrases containing the single word IT. The definition for the PRONOUN feature was expanded for MUC-6.

There are many non-anaphoric uses of the pronouns IT and ITS, i.e., occurrences of these pronouns in contexts in which they do not refer to previously mentioned entities. Several non-anaphoric examples are shown in the following text fragment.

> Asked why he would choose to voluntarily exit while he still is so young, Mr. James says ***it is time*** to be a tad selfish about how he spends his days.
>
> Mr. Dooner, who recently lost 60 pounds over three-and-a-half months, says now that he has "reinvented" himself, he wants to do the same for the agency. For Mr. Dooner, it means maintaining his running and exercise schedule, and for the agency, it means developing more global campaigns that nonetheless reflect local cultures. One McCann account, "I Can't Believe It's Not Butter," a butter substitute, is in 11 countries, for example.
>
> Maybe he'll even leave something from his office for Mr. Dooner. Perhaps a framed page from the New York Times, dated Dec. 8, 1987, showing a year-end chart of the stock market crash earlier that year. Mr. James says he framed it and kept it by his desk as a "personal reminder. ***It can all be gone like that.***"

There was no time available to create a filter to distinguish anaphoric from non-anaphoric occurrences of these pronouns, either for the training material or for the blind test set. Since the strategy for RESOLVE in MUC-6 was to maximize recall at the expense of precision, RESOLVE attempted to resolve all occurrences of the phrases IT and ITS as possible references to organizations.

Personal pronouns were also captured in the extended definition of the PRONOUN feature, in both their nominal and possessive forms. Table 9.2 contains the complete list of pronominal forms that was included in the definition for this feature for MUC-6.

Table 9.3 Key words used in gender identification

Male	HE, HIM, HIS, MR.
Female	SHE, HER, HERS, MRS., MS.

9.2.3.4 SAME-TYPE

Do both phrases refer to the same type of entity?

Possible values: *YES, NO*

The semantic features available to RESOLVE were not very refined. BADGER distinguished organizations from people, and this information was used to compute the SAME-TYPE feature. Unfortunately, the sentence analyzer did not differentiate subclasses of these two broad categories. For example, there was no distinction made between financial institutions and manufacturing companies, nor was a distinction made between political figures and corporate officers. Deeper semantic analysis would have been very useful for the MUC-6 coreference task, though the level of semantic analysis included in BADGER was probably sufficient for the other three tasks.

9.2.3.5 SAME-NUMBER

Do the phrases agree in number?

Possible values: *YES, NO, UNKNOWN*

Multi-referent phrases, i.e., phrases that refer to more than one entity, were not evaluated when RESOLVE was trained and tested on the MUC-5 EJV domain. Relevant multi-referent phrases were rather rare, and the extra infrastructure needed for comparing multi-referent phrases and singular phrases was considerable.

Multi-referent phrases were not ignored for MUC-6. Instead a simple feature was defined to try to identify how many distinct referents were being referenced by a given phrase. If a singular generic organization description, corporate designator, person title or person role was detected, then the number was 1. If a plural generic entity descriptor or person role was detected, then the number was *plural*. Otherwise, if any number word between "two" and "ten" was detected, that number was used. Other numbers were not considered.

9.2.3.6 SAME-GENDER

Do the phrases agree in gender?

Possible values: *YES, NO, UNKNOWN*

A similarly simple approach was taken with regard to personal pronouns. Candidate phrases were searched for obvious gender information; Table 9.3 shows the words that could be used to identify phrases as referring to males or females.

9.2.3.7 MOST-RECENT-COMPATIBLE

Is phrase 1 the most recent phrase that is compatible with phrase 2?

Possible values: *YES, NO*

As was noted earlier, pronouns did not figure prominently in the MUC-5 EJV domain, and those that occurred in relevant contexts were almost always easily identified as references to joint venture children.

Since *all* pronouns were potential candidates in the MUC-6 coreference task, additional knowledge needed to be included in the feature set used by RESOLVE. Since development time was limited, as was the depth of semantic analysis performed by BADGER, a set of related heuristics was hastily crafted to try to capture some of the pronominal coreference links in the corpus.

Many pronouns can be resolved by simply searching backward in the text for the most recently occurring phrase that is compatible with the pronoun in terms of type, number and gender. Unfortunately, the information available regarding these three aspects of a candidate phrase was rather sparse. There were only two types: persons and organizations (see Section 9.2.3.4, and only the most obvious indications of number and gender were captured (see Sections 9.2.3.5 and 9.2.3.6).

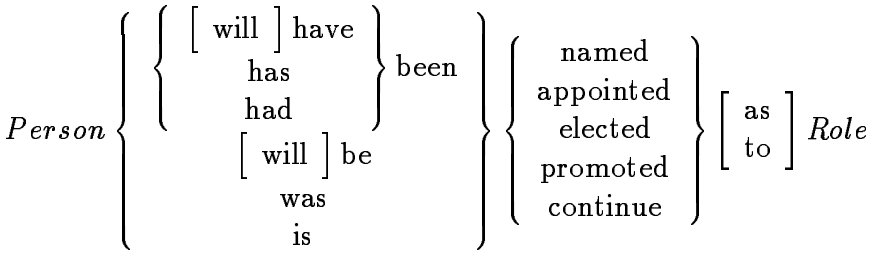
Four different variations were tried on this simple strategy of searching backward for compatible phrases:

- MOST-RECENT-COMPATIBLE: This feature has the value *YES* when phrase 1 is the closest, preceding, compatible candidate – in terms of type, number and gender – to phrase 2; otherwise it has the value *NO*.
- MOST-RECENT-COMPATIBLE-NAMED: This feature has the same definition as the MOST-RECENT-COMPATIBLE feature, except that phrase 1 has the additional restriction that it must contain a *NAME*.
- MOST-RECENT-COMPATIBLE-SUBJECT: This feature has the same definition as the MOST-RECENT-COMPATIBLE feature, except that phrase 1 has the additional restriction that it must have occurred as a *SUBJECT*.
- MOST-RECENT-COMPATIBLE-NAMED-SUBJECT: This feature represents a combination of the two previous variations: it has the same definition as the MOST-RECENT-COMPATIBLE feature, except that phrase 1 has the restrictions that it must both contain a *NAME* and have occurred as a *SUBJECT*.

9.2.3.8 PERSON-IS-ROLE

Possible values: *YES, NO*

Some common variations of predicate nominatives are captured in the feature X-IS-Y (see Section 8.2.6.2). Since the MUC-6 domain focused on stories describing corporate management changes, there were many variations of expressing the event of a person assuming a new role. A new, domain-specific feature was created to capture some of the variations that were noticed in the training materials:



9.2.4 Coreference Results in MUC-6

A set of 30 texts was provided for a "dry run" for MUC-6, which was used to ensure that participating systems were able to accept the format of the texts, and able to generate output in the correct format. Although the UMass MUC-6 system did not participate in the dry run, these texts were available to us.

A preliminary evaluation of RESOLVE – in which the system was trained on the instances generated from the 25 texts annotated with CMI, and tested on instances generated from the output of BADGER processing the 30 "blind" texts used in the MUC-6 "dry run" – produced results for which recall was 30% and precision was 47%. This preliminary evaluation was conducted only two days before the official final evaluation run.

The final two days preceding the final evaluation were spent on three primary tasks:

- Refining the filters that attempted to eliminate candidate phrases that had been erroneously been tagged as organizations or persons;
- Refining the trimming functions that attempted to pare back candidate phrases that had been extended too far by BADGER's noun phrase analysis module;
- Adding a capability to split apart conjunctions and appositives – which were not split prior to classification by RESOLVE – in order to generate system output that more closely matched the guidelines for handling these constructs.

These late efforts appear to have increased our score, even though they did not have much to do with the specific task of coreference resolution (only the preprocessing and post-processing of candidate phrases). Our official score for the MUC-6 coreference task was 44% recall and 51% precision.[3]

9.3 The Discovery of a Domain-Specific Rule

The official MUC-6 Coreference Task results for RESOLVE were better than we anticipated. However, the real surprise in the application of RESOLVE to the MUC-6 coreference task was that it learned how to classify coreferent phrases in contexts for which it had no explicit domain-specific knowledge.

Figure 9.1 illustrates one branch of the decision tree used by RESOLVE in the MUC-6 coreference task. This branch of the decision tree can be described more

[3]Other MUC-6 recall scores ranged from 36% to 63%; other precision scores ranged from 44% to 72%. Note that the system that achieved the highest recall score was not the same system that achieved the highest precision score.

```
ALIAS = NO
 SAME-TYPE = YES
  PRONOUN-2 = NO
   ...
    SAME-NUMBER = UNKNOWN
     ...
      PRONOUN-1 = NO
       NAME-2 = NO
        ...
           SAME-SENTENCE = YES
            NAME-1 = YES: "+"
```

Figure 9.1 One branch of RESOLVE's MUC-6 decision tree

IF	*both phrases are the* ***same type***
AND	*neither phrase is a* ***pronoun***
AND	*the first phrase includes a* ***name***
AND	*the second phrase does not include a* ***name***
AND	*both phrases are in the* ***same sentence***
THEN	class = **YES** (the phrases are coreferent)

Figure 9.2 A rule corresponding to the MUC-6 tree branch in Figure 9.1

compactly by the rule in Figure 9.2 (which assumes that a previous rule has checked for ALIAS).

This rule may seem unintuitive and risky. It is questionable whether anyone who was manually constructing a rulebase to classify coreferent phrases would have even thought of this rule. Furthermore, if this rule were explicitly suggested to someone (without providing many examples of its application), it may well have been rejected.

Despite these potential drawbacks, however, this rule turned out to be extremely useful. The rule was discovered during preparation for the MUC-6 conference, in an analysis of what RESOLVE did in the walk-through text selected as the focus of presentations at the conference.[4]

In the MUC-6 walk-through text, this rule was applied to 14 pairs of phrases: it correctly classified eight of these instances, and the remaining six instances all contained semantic tagging errors. Section C.8.2 includes the complete sentences for each of the rule applications; the applications of this rule to the sample text provided in Section C.2 can be found in Section C.8.1.

Two of the correctly classified instances in the walk-through text were composed of pairs of phrases that had been incorrectly separated by the appositive classifier. One instance included a proper noun and a reflexive pronoun (no features explicitly

[4]The MUC-6 walk-through text is rather long, so a shorter text was selected as a sample text for MUC-6 – see Section C.2. However, selected sentences from the walk-through text can be found in Section C.8.2.

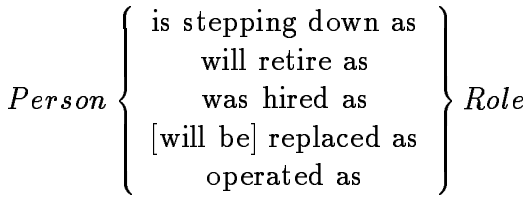


Figure 9.3 New PERSON-IS-ROLE patterns covered by the rule

Table 9.4 Applications of the rule in the MUC-6 final evaluation corpus

Description	*Count*
Correct	41
Semantic errors	19
Misidentified plurals	18
Missed phrases/constructs	5
Indefinite expressions	10
Other errors	9
Total	102

captured knowledge focusing on reflexive pronouns). The other five instances were drawn from contexts in which a person was starting or leaving a position, as shown in Figure 9.3.

While there was an explicit feature PERSON-IS-ROLE (see Section 9.2.3.8) that was developed for the MUC-6 domain, the definition of this feature was based on patterns that were seen in the 25 training texts. There were only five main verbs included in the PERSON-IS-ROLE pattern, and this single text illustrates five others. There are undoubtedly many more verbs that are used to express relationships between people and roles throughout the rest of the corpus.

It would be difficult to list all the verbs that might be used in expressions that link people and roles. Fortunately, an exhaustive list was not necessary for some of these expressions, since RESOLVE learned a rule that captured many of them.

The pattern captured by the rule in Figure 9.2 did not occur in the decision trees trained on the EJV domain, and it may not be a useful pattern in other domains. However, it turned out to be quite useful in the MUC-6 walk-through text: if the upstream semantic tagging errors were eliminated, this branch of RESOLVE's decision tree was correct 100% of the time.

The rule was used 102 times throughout the 30 texts that comprised the final evaluation corpus for the MUC-6 coreference task. A breakdown of these applications is shown in Table 9.4. The discovered rule was applied correctly 41 times (though 6 of these applications involved phrases that were not properly trimmed). Most of the erroneous applications of this rule were the result of a variety of upstream errors: semantic tagging (19), misidentified plural expressions (18), and intervening phrases

or constructs that were missed entirely by the sentence analyzer (5). There were 10 misapplications of the rule to pairs of phrases in which the second (later) phrase begins with an indefinite article or occurs in a context that suggests a possibility rather than a fact: the MUC-6 coreference task guidelines specify that indefinite expressions and expressions of possible positions were not to be considered candidates for coreference resolution. Unfortunately, there was a bug in the feature definition for INDEF-ART-2 with the result that its value was always *NO* in training, so this feature was never included in the decision tree.[5] There were 9 misapplications of the rule that cannot be explained by errors in the upstream processing of the phrases.

The perceived effectiveness of this rule depends on which errors are counted and which are ignored for the purpose of evaluating coreference resolution as a distinct task. The first three categories of errors – semantic tagging, misidentified plurals and missed phrases and constructs – are certainly not attributable to RESOLVE; the fourth category is attributable to RESOLVE, but might be considered a "dumb bug". If only the final category of "other errors" is considered, then this particular rule in RESOLVE was correct in 40 out of 49 – or approximately 82% – of its applications.

One of the reasons that this rule is so useful is it compensates for the fact that RESOLVE did not have access to a more extensive knowledge base, for example, the type of knowledge that would have identified a much large number of verb phrases that express the concept that a person is accepting or leaving a position. Part of the reason for this lack of knowledge was the time constraints – very little time was available for engineering more domain-specific knowledge, particularly knowledge that would be useful for coreference resolution.

However, these knowledge engineering issues highlight one of the benefits of the using machine learning for coreference resolution: it is unlikely that *any* system could have complete and comprehensive knowledge of all the ways of describing an event wherein a person accepts or leaves a position. The ability of RESOLVE to compensate for incomplete knowledge may be useful even for information extraction systems with more knowledge.

9.4 Using RESOLVE for Other MUC-6 Tasks

Due to the severe time limitations imposed during the MUC-6 evaluation, and the dependencies among different system modules, the training material collected from the MUC-6 texts via CMI were not yet available when a final version of the coreference module was required. Therefore, in its application to the TE and ST tasks, RESOLVE was trained on the annotations collected from the MUC-5 EJV domain; however, only a subset of the domain-independent features – those for which information was available from other components in BADGER – were used.

Table 9.5 lists the set of features that were used for the training and testing of RESOLVE on these two tasks.[6] Due to the nature of the TE and ST evaluations, a straightforward evaluation of RESOLVE's performance as a subcomponent is not possible. The important aspect to note about this application of RESOLVE is that it represents the fulfillment of one of the intents of its designer – it has been successfully used as the coreference resolution module in an information extraction system.

[5] This aspect of RESOLVE's performance in MUC-6 is further complicated by a misinterpretation of the guidelines on the part of the annotator: indefinite expressions *were* considered candidates for coreference in the preparation of the training material.

[6] See Section 8.2 for a description of each of these features.

Table 9.5 Features used in the MUC-6 TE/ST Version of RESOLVE

DEF-ART-i
INDEF-ART-i
PRONOUN-i
LOC-i
GOVERNMENT-i
NAME-i
SAME-STRING
SUB-STRING
SAME-SENTENCE
PREVIOUS-SENTENCE
SAME-SENTENCE
PREVIOUS-SENTENCE
SAME-CONSTITUENT
BOTH-SUBJECT
BOTH-GOVERNMENT
COMMON-HEAD-NOUN
COMMON-MODIFIER
COMMON-HEAD-NOUN/MODIFIER
COMMON-NP
COMMON-LOC
ALIAS
X-SAID-IT
X-IS-Y

CHAPTER 10

CONCLUSIONS

This dissertation represents an exploration into the use of machine learning techniques for coreference resolution within the context of an information extraction system. In particular, it describes RESOLVE, a system that learns how to classify pairs of phrases as coreferent or not coreferent.

One of the motivations in applying machine learning to the coreference resolution task is the complexity of the coreference resolution problem. There are a number of different types of knowledge that human readers bring to bear in determining coreferent relationships among phrases in a text. Simply identifying these knowledge sources, or features, is a challenging task; combining and ordering these features to achieve the best possible performance is a considerably more daunting task. Once a combination of features – as represented by a decision tree – has been determined for coreference resolution in one particular domain, that particular combination may or may not be suitable for a different domain.

A machine learning algorithm makes decisions about how to combine features in order to solve a given problem. The issue then becomes how to structure a given problem in order to permit a machine learning algorithm to be used. This dissertation has presented a detailed description of many of the design decisions that were made in order to create a framework in which machine learning techniques could be applied to the coreference resolution problem.

The primary contribution of this dissertation, however, is that it has shown that a machine learning approach to coreference resolution is not only *possible* but *effective*, i.e., there are a number of benefits to using machine learning techniques for coreference resolution. The remainder of this chapter will highlight some of the benefits that have been discussed in previous chapters, and will conclude with a description of some potential directions for extending this work.

10.1 Principal Claims

Chapter 1 presented an organization of the work described in this dissertation that is based around two primary research contributions:

- RESOLVE demonstrates that a machine learning approach to coreference resolution can achieve the same level of performance as a manually engineered approach, but with less human effort required.
- RESOLVE shows that domain-specific knowledge is important for coreference resolution performance.

Each of these primary contributions will be reviewed in greater detail in the following sections.

10.1.1 The Efficacy of a Machine Learning Approach

Chapter 7 describes a set of experiments in which the performance of rules used in the manually-engineered coreference module of an information extraction system is compared to the performance of RESOLVE. The data used for the evaluation of both systems was collected from the MUC-5 EJV corpus, the domain for which the rule-based system was originally developed.

When RESOLVE has access to the same set of features that were used in the antecedents of the rules, it achieves the same level of performance as the rule-based system. The decision tree created by RESOLVE contains an arrangement of features that differs from the arrangement used in the rule-based system. However, both systems achieve levels of performance that are statistically indistinguishable, both in terms of recall and precision. The advantage of using RESOLVE is that the level of effort is considerably less than is required for manually engineering a rule-based system: the individual features must still be defined manually, but the combinations and ordering of features is done by a machine learning algorithm.

One of the features used by both RESOLVE and the rule-based system determines whether two phrases both refer to joint venture companies (BOTH-JV-CHILD); another feature determines whether exactly one of the two phrases refers to a joint venture company (XOR-JV-CHILD). These *meta-features* are defined in terms of more primitive features that determine whether the first phrase refers to a joint venture company (JV-CHILD-1) and whether the second phrase refers to a joint venture company (JV-CHILD-2). When RESOLVE has access to only the primitive features, i.e., the meta-features are not available for training or testing, the concepts represented by the meta-features are still represented in the decision trees as subtrees containing combinations of the primitive features.[1] Furthermore, the performance of RESOLVE does not degrade when it does not have access to these meta-features.

The C4.5 machine learning algorithm has a special method for constructing a decision tree based on instances that have some features whose values are unknown. Some of the features used in these experiments have *unknown values* in a large proportion of the instances. When decisions are made by C4.5 based on the distribution of known values for such a feature, decision trees that represent impossible combinations of features can result. If, instead, *UNKNOWN* is treated as a *first class value*, e.g., a third possible value in addition to *YES* and *NO*, the decision trees do not contain any impossible combinations of features. Perhaps more importantly, decision trees that have been trained with instances in which *UNKNOWN* is a first-class value achieve higher performance than those trained with normal handling of unknown values or those trained with instances that do not contain *UNKNOWN* values. In fact, the decision trees trained with first-class *UNKNOWN* values achieve performance that is better than the rule-based system, which did not make explicit use of *UNKNOWN* values.

10.1.2 The Importance of Domain-Specific Knowledge

The set of features used for the experiments reported in Chapter 7 is not very extensive – this set was constrained by the knowledge used by the rule-based system to which RESOLVE was being compared. More knowledge, in the form of a broader set of features, was made available to RESOLVE for the experiments reported in Chapter

[1] Although, as noted in Section 7.2.4.2, only a partial representation of the XOR-JV-CHILD concept appears in the decision trees. However, this partial representation was able to capture the most salient aspects of the concept, since the performance did not suffer by not having the full representation of the XOR-JV-CHILD concept in the decision tree.

8. As might be expected, RESOLVE achieves higher levels of performance, especially in terms of recall, when it has access to more knowledge.

Many of the new features added to RESOLVE were domain-independent: they encoded knowledge about pronouns, definite articles, and syntax. However, a few of the features that were added encoded domain-specific knowledge: knowledge required for identifying the *parents* of joint venture companies (JV-PARENT-i) not just the joint venture companies (JV-CHILD-i) themselves.

This broader set of features can be partitioned into two disjoint subsets: a domain-independent set of features and a domain-specific set of features. The importance of domain-independent knowledge in comparison to domain-specific knowledge was be assessed by building a two systems – one system with access to all the available knowledge and another system with access to only the domain-independent knowledge – and comparing their performance. The construction of two different systems using a machine learning algorithm is easier than constructing two different systems manually, another benefit of using machine learning techniques for coreference resolution.

When RESOLVE is given access only to the domain-independent features for training and testing, i.e., the eight domain-specific features are disabled, it achieves only 80% of the recall as it does when it is given access to all of the features (domain-independent and domain-specific). As a control condition, when any 8 domain-independent features are disabled, the system still achieves over 96% of the recall that it achieves when all features are available to it. Thus, the domain-specific features have more impact on the system's performance than any similarly sized set of domain-independent features.

10.2 Other Contributions

The description of RESOLVE in this dissertation was organized around the two main claims outlined above. In addition to these contributions, there are a number of other contributions made by this work. These other contributions are the focus of this section.

10.2.1 RESOLVE as an Information Extraction System Component

Chapter 9 focused on the use of RESOLVE in the MUC-6 Coreference Task evaluation. In fact, throughout this dissertation, the use of RESOLVE as a stand-alone coreference resolution system has been emphasized, since this was the mode which makes evaluation of coreference resolution performance possible. However, RESOLVE was also used as the coreference resolution component of a larger system – the BADGER information extraction system that was used for the MUC-6 Template Element (TE) and Scenario Template (ST) tasks [Fisher *et al.*, 1995] – thereby achieving one of the goals behind the development of this new approach to coreference resolution.

Although RESOLVE was given very little domain-specific knowledge for this new domain, it was able to discover domain-specific combinations of domain-independent features, acquiring knowledge for coreference resolution in a new domain that is comprehensible, interesting and useful (for that domain). One such "learned rule" was illustrated in Chapter 9, a rule that links people with their roles in a company (an important relationship for the MUC-6 task). A special feature (PERSON-IS-ROLE) was defined for this kind of coreference relationship, but it was too narrowly defined. The learned rule helps to compensate for this overly constrained feature, and although it may appear to be both unintuitive and risky, a trace of its application in 30 texts shows that it is a useful rule.

10.2.2 A Successful Application of ML to NLP

Many issues involved in applying machine learning techniques to a problem in natural language processing were discussed in Chapter 4. Most of the NLP applications of ML have been at the level of sentence analysis, e.g., part-of-speech tagging and prepositional phrase attachment. RESOLVE represents a case study amid a growing collection of successful ML/NLP applications at the level of discourse analysis, i.e., linguistic phenomena that cross sentence boundaries.

It is hoped that some of the details of this application of ML techniques to an NLP problem – particularly at the discourse level – might serve as a guide to other researchers who explore the possibility of using ML techniques to their NLP problems.

10.2.3 New Data Sets for Machine Learning

Two data sets were collected within the context of the work described in this dissertation – a set of annotations from 50 texts in the MUC-5 EJV domain and another set of annotations from 25 texts in the MUC-6 domain of corporate management changes. The instances that were generated from these annotations have already been shared with a number of other researchers – some with a background in natural language processing and others with a background in machine learning. These data sets may be made available to other researchers as well.

10.3 Future Work

The work described in this dissertation represents an initial investigation into the application of machine learning techniques to the problem of coreference resolution for an information extraction system. In addition to the contributions made by this work, enumerated in the previous section, a number of further contributions could be made by extending this work in new directions. Some of these potential extensions are discussed below.

10.3.1 Better Selection of Plausible Alternatives

The current system selects the first positive instance found when establishing coreference links between phrases. If more than one previous reference is considered coreferent with a new reference, i.e., if a reference has more than one positive classification associated with it, a more effective procedure could be developed to select among these alternatives.

The instance representation selected for the work described in this dissertation was to pair individual phrases, one phrase being a newly encountered phrase, the other being a previously encountered phrase. Another option, discussed in Section 4.1.1.3, would have been to group all preceding coreferent phrases, so that rather than comparing a new phrase to a single preceding phrase, a new phrase would be compared to a group of preceding phrases (all of which co-refer).

The advantage of the latter approach is that all the information available about previous referents is kept together, so if the new phrase contains a name, it may be an alias of one of the preceding phrases already associated with the group, or if the new phrase contains location information, it may be a location that is compatible with a location mentioned in one of the preceding phrases associated with the group. The primary disadvantage, described in greater detail in Section 4.1.1.3, is that such an approach complicates evaluation.

If the new phrase is a potential match on three previous phrases, and two of those phrases have already been deemed coreferent with each other, then that information could be used during classification to bias the system to choose one of those two, coreferent, previous phrases.

10.3.2 Training on a Subset of the Positive Instances

The instances generated for the experiments described in this dissertation included *all* pairwise combinations of coreferent phrases. Examination of these pairings reveals that some of these positive instances may be quite difficult to classify for a human coreference resolution system.

For example, in one text, the following references were made to an *entity*:

1. `DAIWA BANK`
2. `THE JAPANESE BANK`
3. `DAIWA`

Since Daiwa is the only bank mentioned in the text, the link between references 1 and 2 is straightforward; the third reference is a simple alias of the first reference. However, the link between the second and third references is difficult to establish – for a human or machine – without relying on the transitive closure of coreference.

The use of instances representing the pairing of references whose coreference relation is difficult to determine by a human reader may serve only to confuse a machine learning system – the learning system will attempt to capture such coreference relationships with somewhat arbitrary combinations of features (since no meaningful combination of features is possible).

Unfortunately, selecting only the *best* pairings of coreferent phrases (i.e., positive instances) for training would reduce the size of the training set, and further skew the balance between positive and negative instances. However, the resulting representation of the learned concept is much more likely to be more compact and more understandable, since there are fewer "strange" cases to account for. Furthermore, the classification performance ought to be improve, since the "easy" cases are more likely to be correctly classified, and transitive closure will be more likely to account for the difficult coreference links.

10.3.3 New Pruning Procedures Based on Recall and Precision

The goal of C4.5, and many other machine learning algorithms, is to maximize accuracy (or, conversely, to minimize errors). This goal drives the pruning procedure in C4.5, i.e., the procedure attempts to simplify a decision tree so that it has the highest possible accuracy (or lowest possible error).

Measuring the *accuracy* of a coreference resolution system is not as informative as measuring its *recall* and *precision*.[2] Unfortunately, there is a fundamental tension between recall and precision such that maximizing recall often results in lower precision and maximizing precision often results in lower recall.[3]

For information extraction systems, as with information retrieval systems, different users may have different requirements with respect to each of these two metrics

[2] See Section 6.1.1.

[3] See Section 6.2.1.

– some people may be more willing to sacrifice some precision to attain high recall, while others will give up some recall in order to attain high precision.[4]

A pruning procedure that takes into account the different classes of errors that affect recall and precision[5] may be more appropriate than one that simply focuses on the overall error rate. Pazzani *et al.* [1994], propose a scheme for a machine learning algorithm to incorporate information about the relative costs of different classes of errors as it constructs a decision list. A table representing different costs to be associated with false positive errors and false negative errors could be used to bias such a system toward higher recall or higher precision.

Another approach to incorporating a high recall or high precision bias into a machine learning algorithm would be to modify the C4.5 pruning procedure. One of the sources of information used by the pruning procedure uses is an estimate of errors; a weighting factor could be added to the error estimating function, controlled by an additional parameter to C4.5. This parameter could indicate the relative cost of false positive errors versus false negative errors. High values of this parameter would penalize false positive errors more heavily than false negative errors, leading to a high precision bias; low values of this parameter would penalize false negative errors more heavily than false positive errors, leading to a high recall bias.

10.3.4 Other Machine Learning Algorithms

C4.5 was selected on the basis of its comprehensibility, availability and widespread use (see Section 4.2). The focus of this dissertation has been on the development of features for coreference resolution rather than on a comparison of different machine learning algorithms or different algorithm parameters for this task. The previous section discussed some possible extensions based on tuning C4.5 for this task; this section will present some potential extensions that involve other learning algorithms.

The features used by RESOLVE are not biased toward any one particular theory of coreference resolution, although many of these features are elements of previously proposed theories of coreference resolution. RESOLVE makes use of a small set of meta-features, i.e., features defined in terms of other features; this process of incorporating additional knowledge – manually combining features that may or may not be combined by the learning algorithm – could be carried a step further by combining larger sets of features, perhaps representing entire theories of coreference resolution.

For example, one "feature" might represent a pronoun resolution algorithm proposed in previous research; if this algorithm is useful for a particular domain, the learning algorithm will incorporate this feature into its concept representation for that domain. Ortega and Fisher [1995] have presented a way of integrating a domain theory with training data (which may conflict with some aspects of the domain theory) in such a way that the best elements of both the theory and the data can be incorporated into a classifier. This approach may be applied to the task of coreference resolution, and may even provide a mechanism for evaluating how well different theories of coreference resolution – or more specific theories for, say, pronoun resolution – cope with the types of coreferent relationships that are found in a new domain.

[4] Of course, all users would really like high recall ***and*** high precision; however some users may place more relative importance on one factor or the other.

[5] As noted in Section 6.2.2, recall is inversely related to the false negative error rate (fewer false negatives tends to produce higher recall scores) and precision is inversely related to the false positive error rate (fewer false positives tends to produce higher precision).

10.3.5 Coreference Resolution and Information Extraction

Many information extraction system developers who have attempted a solution at coreference resolution have recognized the inherent bias in their implementation. This bias results in a tendency toward either *lazy merging* or *under-merging*, where too little merging is done (reflecting a bias toward classifying a pair of references as not coreferent), or *greedy merging* (or *over-merging*), where too much merging is done (reflecting a bias toward classifying a pair of references as coreferent) [Hirschman, 1992].

Unfortunately, few information extraction system descriptions contain any mention of this bias, though there seems to be an implicit assumption among MUC participants that lazy merging is less dangerous than greedy merging. This assumption is not only unstated but appears to be untested. If tighter control can be exerted over RESOLVE's recall and precision, as outlined above in Section 10.3.3, it will become possible to test whether lazy merging or greedy merging is a more effective bias for an information extraction system.

APPENDIX A

EARLY MUCS

This chapter will provide some background on the first four Message Understanding Conferences. The first two conferences were primarily organizational – deciding what and how to evaluate performance on an information extraction task. The most recent four conferences share a more common and standardized structure: MUC-3 and MUC-4 will be described in this chapter; MUC-5 and MUC-6 will be described separately in separate chapters, since much of the work described in this dissertation is based on these tasks.

A.1 MUCK-I and MUCK-II

The first Message Understanding Conference (MUCK-I) was a forum in which different NLP systems were used to analyze a set of 12 teletype-style texts describing naval tactical operations [Sundheim and Dillard, 1987]. The main focus of this first conference was to evaluate whether NLP system developers could extend their existing systems, *during* the conference, to analyze texts in this new domain. Since there was no standardized output format, the evaluation was necessarily qualitative rather than quantitative [Sundheim, 1989, page 1].

The corpus of naval tactical operations messages was expanded from 12 to 130 for the Second Message Understanging Conference (MUCK-II), and a standardized output format was created to represent the information extracted from any given message [Sundheim, 1989]. The larger corpus enabled more extensive evaluation of the participating systems and the standardized output permitted a quantification of system performance.

The subsequent Message Understanding Conferences all had a set of common features:

- A corpus comprised of a set of *texts* and a set of corresponding *key templates* which encoded the information that should extracted from each text; each participating system was to generate *response templates* which contained the information that actually was extracted from each text by the system.
- The corpus was partitioned into a large *development* or *training* set, which was distributed to the participants well in advance of the official evaluation for development and internal testing purposes, and a *blind* set which was reserved for the official evaluation.
- The official set of blind texts was made available to participants for a one-week official evaluation period approximately one month prior to the conference, during which time the systems were to be run on the texts, generating response templates, without any screening or intervention on the part of the system developers.

- The response templates were compared with the key templates for each text in the blind set using specially created evaluation software. This scoring software computed the *recall* – the percentage of information in a text that is [correctly] extracted by a system – and *precision* – the percentage of information extracted by a system that is correct – for each participating system.[1]

A.2 MUC-3: Latin American Terrorism

Newswire stories in the domain of Latin American terrorism (LAT) formed the corpus used for the Third Message Understanding Conference (MUC-3) [MUC-3, 1991, Sundheim, 1991]. A story relevant to the MUC-3 task would describe one or more specific incidents of either terrorism or state-sponsored violence in one of nine Latin American countries. For each such incident, a system was required to extract information concerning the date, location, weapon(s) used, perpetrator(s), victim(s) and physical target(s), and output this information with appropriate labels in a *response template*. A *key template* was created by a human reader to represent the information that a system should extract from the story. Examples of a MUC-3 sample text and its corresponding key template are included in later sections.

A.2.1 MUC-3 Overview

A corpus of 1300 development (DEV) texts and key templates was distributed to each of the 15 sites participating in MUC-3 for development and testing prior to the final evaluation. Another set of 100 texts, TST1, was used for a preliminary evaluation (a "dry run") conducted several months prior to the official evaluation, and an additional set of 100 texts, TST2, was used for the final evaluation. Both evaluation sets were withheld from partipants until the evaluation dates, and were processed blind by the participants' systems. All three sets of texts were drawn from the same source – newswire reports concerning Latin American Terrorism.

The response templates generated by each participating system were scored against the key templates using special software provided by the sponsors of the evaluation. Many different aspects of system performance were measured, but two of the most important evaluation metrics were recall and precision. *Recall* measures the amount of information [correctly] extracted from a text; in the context of the MUC-3 evaluation, this was computed as the percentage of information in the key templates that was found in the response templates. *Precision* measures how much of the information extracted from a text is correct; in the MUC-3 context, this was computed as the percentage of information in the response templates that is found in the key templates.[2]

[1] In the context of the MUC evaluations, ***recall*** was computed as the percentage of information in the key templates that was found in the response templates and ***precision*** was computed as the percentage of information in the response templates that is found in the key templates. As a simplified example, suppose a key template contains items {A,B,C,D} and a response template contains items {A,B,C,E,F}: since the response template contains 3 of the 4 items in the key template, recall is 75%; since 3 of the 5 items in the response template are also in the key template, precision is 60%.

[2] As a simplified example, suppose a key template contains items {A,B,C,D} and a response template contains items {A,B,C,E,F}: since the response template contains 3 of the 4 items in the key template, recall is 75%; since 3 of the 5 items in the response template are also in the key template, precision is 60%.

A.2.2 Sample MUC-3 Text

TST1-MUC3-0075

BOGOTA, 28 JUL 89 (INRAVISION TELEVISION CADENA 1) -- [REPORT] [MARIBEL OSORIO] [TEXT] A COLOMBIAN JUDGE HAS PAID THE HIGHEST PRICE FOR DOING HER DUTY. MARIA ELENA DIAZ PEREZ, THIRD JUDGE OF PUBLIC ORDER, AND TWO OF HER BODYGUARDS FROM THE DAS [ADMINISTRATIVE DEPARTMENT OF SECURITY], WERE ASSASSINATED IN MEDELLIN TODAY BY A GROUP OF 10 PAID ASSASSINS IN TWO CARS, A MAZDA AND WHAT WAS THOUGHT TO BE A MERCURY.

SHE WAS TRAVELING HOME IN HER CAR TODAY AT NOON. ABOUT HALF A BLOCK FROM HER HOME, HER CAR, A TOYOTA WITH LICENSE PLATE NO. A-3037, WAS INTERCEPTED BY A WHITE MAZDA AND POSSIBLY A MERCURY. DIAZ' DRIVER HAD TO HIT THE BRAKES, AS THE MOTORCYCLE OF ONE OF HER BODYGUARDS SLAMMED INTO THE REAR OF HER CAR. IN A MATTER OF SECONDS, THE 10 PAID ASSASSINS OPENED FIRE ON THE JUDGE'S CAR. A TOTAL OF 55 9-MM SUBMACHINE GUN ROUNDS HIT THE LEFT SIDE OF THE CAR. THE JUDGE, WHO WAS SITTING IN THE BACK SEAT, WAS ABLE TO LEAVE THE CAR AND RUN, BUT SHE WAS ONLY ABLE TO TAKE A FEW STEPS BEFORE BEING GUNNED DOWN BY 19 BULLETS. TWO OF HER DAS BODYGUARDS, DAGOBERTO RODRIGUEZ AND ALFONSO DE LIMA, WHO WERE INSIDE THE CAR, WERE KILLED INSTANTLY.

NO ONE HAS BEEN ARRESTED YET IN CONNECTION WITH THIS INCIDENT. THE TRUTH IS THAT THE MANY THREATS MADE AGAINST HER BECAME A REALITY TODAY.

MARIA ELENA DIAZ, 34, WILL BE BURIED IN MEDELLIN TOMORROW.

A.2.3 Sample MUC-3 Key Template

```
0.  MESSAGE ID                     TST1-MUC3-0075
1.  TEMPLATE ID                    1
2.  DATE OF INCIDENT               28 JUL 89
3.  TYPE OF INCIDENT               MURDER
4.  CATEGORY OF INCIDENT           -
5.  PERPETRATOR: ID OF INDIV(S)    "GROUP OF 10 PAID ASSASSINS" /
                                   "10 PAID ASSASSINS"
6.  PERPETRATOR: ID OF ORG(S)      -
7.  PERPETRATOR: CONFIDENCE        -
8.  PHYSICAL TARGET: ID(S)         *
9 . PHYSICAL TARGET: TOTAL NUM     *
10. PHYSICAL TARGET: TYPE(S)       *
11. HUMAN TARGET: ID(S)            "MARIA ELENA DIAZ PEREZ"
                                     ("THIRD JUDGE OF PUBLIC ORDER" /
                                      "JUDGE")
                                   "DAGOBERTO RODRIGUEZ" ("BODYGUARDS")
                                   "ALFONSO DE LIMA" ("BODYGUARDS")
12. HUMAN TARGET: TOTAL NUM        3
13. HUMAN TARGET: TYPE(S)          LEGAL OR JUDICIAL:
                                     "MARIA ELENA DIAZ PEREZ"
                                   SECURITY GUARD:
                                     "DAGOBERTO RODRIGUEZ"
                                   SECURITY GUARD:
                                     "ALFONSO DE LIMA"
14. TARGET: FOREIGN NATION         -
15. INSTRUMENT: TYPE(S)            MACHINE GUN
16. LOCATION OF INCIDENT           COLOMBIA: MEDELLIN (CITY)
17. EFFECT ON PHYSICAL TARGET(S):  *
18. EFFECT ON HUMAN TARGET(S):     *
```

A.3 MUC-4: New Template Structure

The same domain was retained for the Fourth Message Understanding Conference (MUC-4) [MUC-4, 1992]. The primary difference between the two evaluations was a revision to the template definitions; the retention of the same domain permitted an assessment of how much systems had matured over the course of a year – many of the MUC-3 veterans also participated in MUC-4.

A.3.1 MUC-4 Overview

One of the goals for the Fourth Message Understanding Conference (MUC-4) [MUC-4, 1992] was to assess progress among the systems that had participated in MUC-3 [Sundheim, 1992]. A total of 17 sites participated in MUC-4, 12 of which were MUC-3 "veterans". The 1500-text MUC-3 corpus – the DEV, TST1 and TST2 sets – was used as the development corpus for MUC-4. Two additional 100-text sets, TST3 and TST4, were used for the final evaluation. The template definition from MUC-3 was revised in order to achieve a better representation of the relevant information from each incident and to enable more accurate measurement of system performance on the task.[3] As might be expected, after spending another year on development and testing in the same domain, most veteran systems showed significant performance improvement between MUC-3 and MUC-4.

[3] Appendix A.3.2 contains the MUC-4 key template for story TST1-0075.

A.3.2 Sample MUC-4 Key Template

```
0.  MESSAGE: ID                     TST1-MUC4-0075
1.  MESSAGE: TEMPLATE               1
2.  INCIDENT: DATE                  28 JUL 89
3.  INCIDENT: LOCATION              COLOMBIA: MEDELLIN (CITY)
4.  INCIDENT: TYPE                  ATTACK
5.  INCIDENT: STAGE OF EXECUTION    ACCOMPLISHED
6.  INCIDENT: INSTRUMENT ID         "SUBMACHINE GUN" /
                                    "SUBMACHINE GUN ROUNDS" /
                                    "9-MM SUBMACHINE GUN" /
                                    "9-MM SUBMACHINE GUN ROUNDS" /
                                    "55 9-MM SUBMACHINE GUN ROUNDS"
7.  INCIDENT: INSTRUMENT TYPE       MACHINE GUN:
                                      "SUBMACHINE GUN" /
                                      "SUBMACHINE GUN ROUNDS" /
                                      "9-MM SUBMACHINE GUN" /
                                      "9-MM SUBMACHINE GUN ROUNDS" /
                                      "55 9-MM SUBMACHINE GUN ROUNDS"
8.  PERP: INCIDENT CATEGORY         -
9.  PERP: INDIVIDUAL ID             "GROUP OF 10 PAID ASSASSINS" /
                                    "10 PAID ASSASSINS"
10. PERP: ORGANIZATION ID           -
11. PERP: ORGANIZATION CONFIDENCE   -
12. PHYS TGT: ID                    -
13. PHYS TGT: TYPE                  -
14. PHYS TGT: NUMBER                -
15. PHYS TGT: FOREIGN NATION        -
16. PHYS TGT: EFFECT OF INCIDENT    -
17. PHYS TGT: TOTAL NUMBER          -
18. HUM TGT: NAME                   "MARIA ELENA DIAZ PEREZ"
                                    "DAGOBERTO RODRIGUEZ"
                                    "ALFONSO DE LIMA"
19. HUM TGT: DESCRIPTION            "THIRD JUDGE OF PUBLIC ORDER" /
                                    "JUDGE":
                                      "MARIA ELENA DIAZ PEREZ"
                                    "BODYGUARDS":
                                      "DAGOBERTO RODRIGUEZ"
                                    "BODYGUARDS":
                                      "ALFONSO DE LIMA"
20. HUM TGT: TYPE                   LEGAL OR JUDICIAL:
                                      "MARIA ELENA DIAZ PEREZ"
                                    SECURITY GUARD:
                                      "DAGOBERTO RODRIGUEZ"
```

	SECURITY GUARD: "ALFONSO DE LIMA"
21. HUM TGT: NUMBER	1: "MARIA ELENA DIAZ PEREZ" 1: "DAGOBERTO RODRIGUEZ" 1: "ALFONSO DE LIMA"
22. HUM TGT: FOREIGN NATION	-
23. HUM TGT: EFFECT OF INCIDENT	DEATH: "MARIA ELENA DIAZ PEREZ" DEATH: "DAGOBERTO RODRIGUEZ" DEATH: "ALFONSO DE LIMA"
24. HUM TGT: TOTAL NUMBER	- [4]

APPENDIX B

MUC-5

B.1 MUC-5 Overview

An important question that was not addressed in MUC-4 was how well the participating systems would perform in a new domain, i.e., how portable the systems were. In an effort to address this issue, the domain of Latin American Terrorism was abandoned for the Fifth Message Understanding Conference (MUC-5) [MUC-5, 1993, Sundheim, 1993]. Participants in MUC-5 were provided with four distinct corpora that varied along two dimensions: language and domain [Onyshkevych *et al.*, 1993]. Each of these corpora included key templates representing the extractable information from each text. The MUC-5 participants were required to process texts from at least one of the following language/domain pairs:[1]

- *English Joint Ventures (EJV)*: 1000 news stories, in English, concerning business tie-ups.[2] For each tie-up, an IE system was required to extract information about the entities involved in the joint venture, the people associated with these entities, the facilities used or owned by the new company, and the products or services provided by the new company. The extracted information was represented in a new object-oriented format.[3] A sample text and corresponding key template from the MUC-5 EJV corpus can be found in later sections.

- *English MicroElectronics (EME)*: 1000 news stories, in English, concerning the technology involved in microchip fabrication. For each process, an IE system was required to extract information about the entities involved in the process (e.g., developers, manufacturers, distributors, users), the type of fabrication process, and any devices and equipment used in the process. The extracted information was represented in the new object-oriented format.

- *Japanese Joint Ventures (JJV)*: 1000 news stories and associated templates, both in Japanese, concerning business tie-ups.

- *Japanese MicroElectronics (JME)*: 850 news stories and associated templates, both in Japanese, concerning microchip fabrication processes.

[1] Of the nineteen participating systems, three were run on all four language/domain pairs; two were run on one domain but in both languages (EJV and JJV); one was run on both English domains (EJV and EME). Altogether, 13 systems were run on the EJV domain, seven were run on EME, five were run on JJV and four were run on JME.

[2] A *tie-up* is a relationship among two or more entities (companies, governments, and/or people) created to achieve some business goal, such as marketing or producing a product in a new country.

[3] This new object-oriented template format provided an *efficient* representation of the information [Krupka and Rau, 1992]: entities, people, facilities and products and services are represented as objects, and relationships among these objects are represented as pointers. Unfortunately, the increase in representational efficiency was achieved at the cost of a decrease in *readability*.

The TIPSTER Text Program (Phase I) [TIPSTER, 1993] was an ARPA-funded text-processing program that supported both data detection system development and data extraction system development.[4] The goal of this program was to have a single system that could first detect which documents were relevant to a particular task and then automatically extract the relevant information from the set of relevant documents. Four development teams from each of these two areas were supported by ARPA, many of which included groups from more than one site. The final evaluation for IE systems funded by this program coincided with the MUC-5 evaluation.

Excluding TIPSTER-funded sites, all sites participating in MUC-5 had to select a single language and a single domain in which to run their IE systems. TIPSTER-funded sites, however, had to run their systems in each corpus (EJV, EME, JJV, JME).[5] Thus TIPSTER sites had much more incentive to develop modular IE systems that could be ported easily to new domains as well as to new languages.

B.2 A Note on Evaluation

The metrics used in the MUC and TIPSTER evaluations are evolving [Chinchor, 1991, Chinchor, 1992, Chinchor and Sundheim, 1993]. Refinements are made following each evaluation: the template design is revised to facilitate more accurate measurement of system performance; new aspects of system performance are measured; changes are made to the specification for how some aspects are measured; new measures are created in an effort to summarize overall system performance. As with any evolving system, there will always be some level of error resulting from the evaluation methodology itself (rather than from the performance of a particular system). Human errors in the coding of the key templates introduce additional errors into the evaluation process [Will, 1993].

In spite of the problems with the evolving IE evaluation methodology, the results of the MUC and TIPSTER evaluations still provide a reasonably accurate measurement of system performance on an IE task.[6] This is important not only for assessing the strengths and weaknesses of different approaches (embodied in different systems) to an IE task, but also for assessing the value of different components within a single system. By changing one system component, and holding all others constant, it is possible to measure the contribution of the change made to that component to the overall system performance. As an example, if one were to replace the reference resolution module of a system with a new module, one could assess which module led to better system performance.

[4] Outside of the TIPSTER program, ***document detection*** is often called Information Retrieval (IR) and ***data extraction*** is often called Information Extraction (IE).

[5] UMass, however, joined the two-year project at the midway point, and therefore was required only to develop a system to run in the two English domains.

[6] In fact, the MUC/TIPSTER evaluations provide the ***only*** widely-recognized evaluation framework for assessing IE system performance.

B.3 Sample MUC-5 Text

```
  <doc>

<DOCNO> 0970 </DOCNO>

<DD>    NOVEMBER 30, 1988, WEDNESDAY </DD>

<SO>    Copyright (c) 1988 Jiji Press Ltd.; </SO>

<TXT>

FAMILYMART CO. OF SEIBU SAISON GROUP WILL OPEN A CONVENIENCE STORE IN
TAIPEI FRIDAY IN A JOINT VENTURE WITH TAIWAN'S LARGEST CAR DEALER, THE
COMPANY SAID WEDNESDAY.
THIS WILL BE THE FIRST OVERSEAS STORE TO BE RUN BY A JAPANESE
CONVENIENCE CHAIN STORE OPERATOR.
THE JOINT VENTURE, TAIWAN FAMILYMART CO., IS CAPITALIZED AT 100
MILLION NEW TAIWAN DOLLARS, HELD 51 PCT BY CHINESE AUTOMOBILE CO., 40
PCT BY FAMILYMART AND 9 PCT BY C. ITOH AND CO., A JAPANESE TRADING
HOUSE.
TAIWAN FAMILYMART PLANS TO OPEN SEVEN MORE STORES IN TAIPEI IN
DECEMBER, AND HOPES TO OPEN 200 STORES THROUGHOUT TAIWAN IN THREE
YEARS.
</TXT>
</doc>
```

B.4 Sample MUC-5 Key

<TEMPLATE-0970-1> :=
DOC NR: 0970
DOC DATE: 301188
DOCUMENT SOURCE: `"Jiji Press Ltd."`
CONTENT: *<TIE_UP_RELATIONSHIP-0970-1>*
DATE TEMPLATE COMPLETED: 301192
EXTRACTION TIME: 27
COMMENT: / "TOOL_VERSION: HUME.7.1.3.def"
/ "FILLRULES_VERSION: EJV.7.0.0"

<TIE_UP_RELATIONSHIP-0970-1> :=
TIE-UP STATUS: EXISTING
ENTITY: *<ENTITY-0970-1>*
<ENTITY-0970-2>
<ENTITY-0970-4>
JOINT VENTURE CO: *<ENTITY-0970-3>*
OWNERSHIP: *<OWNERSHIP-0970-1>*
ACTIVITY: *<ACTIVITY-0970-1>*
<ACTIVITY-0970-2>
<ACTIVITY-0970-3>

<ENTITY-0970-1> :=
NAME: `FAMILYMART CO`
ALIASES: `"FAMILYMART"`
NATIONALITY: JAPAN (COUNTRY)
TYPE: COMPANY
ENTITY RELATIONSHIP: *<ENTITY_RELATIONSHIP-0970-1>*
<ENTITY_RELATIONSHIP-0970-2>

<ENTITY-0970-2> :=
NAME: `CHINESE AUTOMOBILE CO`
NATIONALITY: TAIWAN (COUNTRY)
TYPE: COMPANY
ENTITY RELATIONSHIP: *<ENTITY_RELATIONSHIP-0970-2>*
COMMENT: "Taiwan's largest car dealer"

<ENTITY-0970-3> :=
NAME: `TAIWAN FAMILYMART CO`
ALIASES: `"TAIWAN FAMILYMART"`
TYPE: COMPANY
ENTITY RELATIONSHIP: *<ENTITY_RELATIONSHIP-0970-2>*
FACILITY: *<FACILITY-0970-1>*
<FACILITY-0970-2>
<FACILITY-0970-3>

<ENTITY-0970-4> :=
NAME: `C. ITOH AND CO`
TYPE: COMPANY

ENTITY RELATIONSHIP: <*ENTITY_RELATIONSHIP-0970-2*>
COMMENT: "OK to have no entity relationship? only part owner"

<*ENTITY-0970-5*> :=
NAME: `SEIBU SAISON GROUP`
TYPE: COMPANY
ENTITY RELATIONSHIP: <*ENTITY_RELATIONSHIP-0970-1*>

<*FACILITY-0970-1*> :=
LOCATION: TAIPEI (CITY 1) TAIWAN (COUNTRY)
TYPE: STORE

<*FACILITY-0970-2*> :=
LOCATION: TAIPEI (CITY) TAIWAN (COUNTRY)
TYPE: STORE
COMMENT: "seven more stores"

<*FACILITY-0970-3*> :=
LOCATION: TAIWAN (COUNTRY)
TYPE: STORE
COMMENT: "200 stores throughout Taiwan in three years."

<*INDUSTRY-0970-1*> :=
INDUSTRY-TYPE: SALES
PRODUCT/SERVICE: (53 `"CONVENIENCE CHAIN [STORE]"`)
/ (53 `"CONVENIENCE [STORE]"`)

<*ENTITY_RELATIONSHIP-0970-1*> :=
ENTITY1: <*ENTITY-0970-5*>
ENTITY2: <*ENTITY-0970-1*>
REL OF ENTITY2 TO ENTITY1: SUBORDINATE
STATUS: CURRENT

<*ENTITY_RELATIONSHIP-0970-2*> :=
ENTITY1: <*ENTITY-0970-1*>
<*ENTITY-0970-2*>
<*ENTITY-0970-4*>
ENTITY2: <*ENTITY-0970-3*>
REL OF ENTITY2 TO ENTITY1: CHILD
STATUS: CURRENT

<*ACTIVITY-0970-1*> :=
INDUSTRY: <*INDUSTRY-0970-1*>
ACTIVITY-SITE: (<*FACILITY-0970-1*> -)

<*ACTIVITY-0970-2*> :=
INDUSTRY: <*INDUSTRY-0970-1*>
ACTIVITY-SITE: (<*FACILITY-0970-2*> -)
START TIME: <*TIME-0970-1*>

<*ACTIVITY-0970-3*> :=
INDUSTRY: <*INDUSTRY-0970-1*>
ACTIVITY-SITE: (<*FACILITY-0970-3*> -)
START TIME: <*TIME-0970-2*>

<*TIME-0970-1*> :=

DURING: 1288

<*TIME-0970-2*> :=

DURING: 91

COMMENT: "'in three years' = before end of three years.
The fill rules do not adequately cover this situation. LED"

<*OWNERSHIP-0970-1*> :=

OWNED: <*ENTITY-0970-3*>

TOTAL-CAPITALIZATION: 100000000 TWD

OWNERSHIP-%: (<*ENTITY-0970-1*> 40)
(<*ENTITY-0970-2*> 51)
(<*ENTITY-0970-4*> 9)

APPENDIX C

MUC-6

C.1 Overview

The domain chosen for the Sixth Message Understanding Conference (MUC-6) was corporate management changes, e.g., when a corporate officer leaves a position in a company or assumes a position in a[nother] company (or both). Section C.2 contains a sample text from this domain.

The organizers of MUC-6 had two primary goals in constructing the evaluation:

1. *Portability:* encourage development of information extraction systems that are easily retargeted to new domains; and

2. *Accessibility:* encourage broad participation among as many information extraction system developers as possible.

The MUC-6 sponsors and program committee wanted to construct an evaluation that would encourage the development of portable information extraction systems, i.e., systems that could be retargeted to a new domain with little effort. To accomplish this goal, the training materials for the information extraction task were released only one month prior to the final evaluation – in previous evaluations (MUC-3, MUC-4 and MUC-5), training materials had been available for nearly one year prior to the final evaluation.

The MUC-6 program committee also decided to construct an evaluation four different tasks that are important for information extraction – the recognition of proper names in a text, the determination of coreference links among phrases in a text, the merging together of different types of information about a specific entity, and the determination of certain relationships among the entities. Each of the later tasks depends upon the earlier ones; researchers developing information extraction systems – or other NLP applications – but who were unable to field a system to handle the full range of subtasks involved in the task, might still be able to participate in one of the subtasks.

Each of the component tasks is described in more detail in the following sections.

C.1.1 Named Entity Recognition (NE) Task

How well can a system identify proper names referring to people, places and organizations, as well as expressions denoting dates, times, monetary amounts and percentages? All the other tasks in the evaluation depend upon good performance upon this task.

Section C.3 provides an example of the types of information that were to be identified by systems participating in this task.

C.1.2 Coreference (CO) Task

How well can a system determine which nouns and pronouns corefer? Note that the organizers did *not* restrict the candidates for resolution to those phrases that were relevant in some way to the other tasks in MUC-6. All nouns and noun phrases were considered candidates for coreference resolution.

Section C.6 provides an example of the types of information that were to be identified by systems participating in this task.

C.1.3 Template Element (TE) Task

How well can a system merge together all the information relating to individual people and organizations that is contained in a single text? Good identification of names and locations, as well as determination of which references to people and organizations corefer, are necessary for good performance on this task.

Section C.4 provides an example of the types of information that were to be identified by systems participating in this task.

C.1.4 Scenario Template (ST) Task

How well can a system find relationships between people and organizations, given a specific scenario, or definition of "relevant" relationships? For a system to perform well at this task, it would need subcomponents that handled each of the preceding tasks well. This task most closely resembles the task definitions for previous MUC evaluations.

Section C.5 provides an example of the types of information that were to be identified by systems participating in this task.

C.2 Sample MUC-6 Text

<DOC>
<DOCID> wsj93_062.0057 </DOCID>
<DOCNO> 930119-0125. </DOCNO>
<HL> Diller Is Named Chairman,
@ Chief Executive of QVC </HL>
<DD> 01/19/93 </DD>
<SO> WALL STREET JOURNAL (J), PAGE C25 </SO>
<CO> QVCN </CO>
<IN> LIMITED PRODUCT SPECIALTY RETAILERS (OTS),
ALL SPECIALTY RETAILERS (RTS) </IN>
<DATELINE> WEST CHESTER, Pa. </DATELINE>
<TXT>

QVC Network Inc., as expected, named Barry Diller its chairman and
chief executive officer.

Mr. Diller, 50 years old, succeeds Joseph M. Segel, who has been named
to the post of chairman emeritus. Mr. Diller previously was chairman
and chief executive of Fox Inc. and Twentieth Century Fox Film Corp.,
both units of News Corp. He also served for 10 years as chairman and
chief executive of Paramount Pictures Corp., a unit of Paramount
Communications Inc.

Arrow Investments Inc., a corporation controlled by Mr. Diller, in
December agreed to purchase $25 million of QVC stock in a privately
negotiated transaction. At that time, it was announced that Diller
was in talks with the company on becoming its chairman and chief
executive upon Mr. Segel's scheduled retirement this month.

</TXT>
</DOC>

C.3 Sample MUC-6 NE Key

```
<DOC>
<DOCID> wsj93_062.0057 </DOCID>
<DOCNO> 930119-0125.  </DOCNO>
<HL> <ENAMEX TYPE="PERSON">Diller</ENAMEX> Is Named Chairman,
@ Chief Executive of <ENAMEX TYPE="ORGANIZATION">QVC</ENAMEX>
</HL>
<DD> <TIMEX TYPE="DATE">01/19/93</TIMEX> </DD>
<SO> WALL STREET JOURNAL (J), PAGE C25 </SO>
<CO> QVCN </CO>
<IN> LIMITED PRODUCT SPECIALTY RETAILERS (OTS),
ALL SPECIALTY RETAILERS (RTS) </IN>
<DATELINE> <ENAMEX TYPE="LOCATION">WEST CHESTER</ENAMEX>,
<ENAMEX TYPE="LOCATION">Pa.</ENAMEX> </DATELINE>
<TXT>
<p>
<ENAMEX TYPE="ORGANIZATION">QVC Network Inc.</ENAMEX>, as expected,
named <ENAMEX TYPE="PERSON">Barry Diller</ENAMEX> its chairman and
chief executive officer.
</p>
<p>
Mr. <ENAMEX TYPE="PERSON">Diller</ENAMEX>, 50 years old, succeeds
<ENAMEX TYPE="PERSON">Joseph M. Segel</ENAMEX>, who has been named
to the post of chairman emeritus.  Mr. <ENAMEX
TYPE="PERSON">Diller</ENAMEX> previously was chairman and chief
executive of <ENAMEX TYPE="ORGANIZATION">Fox Inc.</ENAMEX> and
<ENAMEX TYPE="ORGANIZATION">Twentieth Century Fox Film
Corp.</ENAMEX>, both units of <ENAMEX TYPE="ORGANIZATION">News
Corp.</ENAMEX> He also served for 10 years as chairman and chief
executive of <ENAMEX TYPE="ORGANIZATION">Paramount Pictures
Corp.</ENAMEX>, a unit of <ENAMEX TYPE="ORGANIZATION">Paramount
Communications Inc.</ENAMEX>
</p>
<p>
<ENAMEX TYPE="ORGANIZATION">Arrow Investments Inc.</ENAMEX>, a
corporation controlled by Mr. <ENAMEX
TYPE="PERSON">Diller</ENAMEX>, in <TIMEX
TYPE="DATE">December</TIMEX> agreed to purchase <NUMEX
TYPE="MONEY">$25 million</NUMEX> of <ENAMEX
TYPE="ORGANIZATION">QVC</ENAMEX> stock in a privately negotiated
transaction.  At that time, it was announced that <ENAMEX
TYPE="PERSON">Diller</ENAMEX> was in talks with the company on
becoming its chairman and chief executive upon Mr. <ENAMEX
TYPE="PERSON">Segel</ENAMEX>'s scheduled retirement this month.
</p>
</TXT>
</DOC>
```

C.4 Sample MUC-6 TE Key

<*ORGANIZATION-9301190125-1*> :=
ORG_NAME: "QVC Network Inc."
ORG_ALIAS: "QVC"
ORG_TYPE: COMPANY
<*ORGANIZATION-9301190125-2*> :=
ORG_NAME: "Fox Inc."
ORG_TYPE: COMPANY
<*ORGANIZATION-9301190125-3*> :=
ORG_NAME: "Twentieth Century Fox Film Corp."
ORG_TYPE: COMPANY
<*ORGANIZATION-9301190125-4*> :=
ORG_NAME: "News Corp."
ORG_TYPE: COMPANY
<*ORGANIZATION-9301190125-5*> :=
ORG_NAME: "Paramount Pictures Corp."
ORG_DESCRIPTOR: "a unit of Paramount Communications Inc."
ORG_TYPE: COMPANY
<*ORGANIZATION-9301190125-6*> :=
ORG_NAME: "Paramount Communications Inc."
ORG_TYPE: COMPANY
<*ORGANIZATION-9301190125-7*> :=
ORG_NAME: "Arrow Investments Inc."
ORG_DESCRIPTOR: "a corporation controlled by Mr. Diller"
ORG_TYPE: COMPANY
<*PERSON-9301190125-1*> :=
PER_NAME: "Barry Diller"
PER_ALIAS: "Diller"
PER_TITLE: "Mr."
<*PERSON-9301190125-2*> :=
PER_NAME: "Joseph M. Segel"
PER_ALIAS: "Segel"
PER_TITLE: "Mr."

C.5 Sample MUC-6 ST Key

<*TEMPLATE-9301190125-1*> :=
DOC_NR: 9301190125
CONTENT: <*SUCCESSION_EVENT-9301190125-1*>
<*SUCCESSION_EVENT-9301190125-2*>
<*SUCCESSION_EVENT-9301190125-3*>
<*SUCCESSION_EVENT-9301190125-4*>
<*SUCCESSION_EVENT-9301190125-5*>
<*SUCCESSION_EVENT-9301190125-6*>
<*SUCCESSION_EVENT-9301190125-7*>

<*SUCCESSION_EVENT-9301190125-1*> :=
SUCCESSION_ORG: <*ORGANIZATION-9301190125-1*>
POST: **"chairman"**
IN_AND_OUT: <*IN_AND_OUT-9301190125-1*>
<*IN_AND_OUT-9301190125-2*>
VACANCY_REASON: REASSIGNMENT
/ DEPART_WORKFORCE
COMMENT: "Segel out, Diller in as chmn of QVC"
/ "Alternative VACANCY_REASON fills: Segel is retiring,
but he is also getting assigned to a new post (emeritus)"
/ "This event could be collapsed with *SUCCESSION_EVENT-2*"

<*SUCCESSION_EVENT-9301190125-2*> :=
SUCCESSION_ORG: <*ORGANIZATION-9301190125-1*>
POST: **"chief executive officer"**
IN_AND_OUT: <*IN_AND_OUT-9301190125-3*>
<*IN_AND_OUT-9301190125-4*>
VACANCY_REASON: REASSIGNMENT
/ DEPART_WORKFORCE
COMMENT: "Segel out, Diller in as CEO of QVC"

<*SUCCESSION_EVENT-9301190125-3*> :=
SUCCESSION_ORG: <*ORGANIZATION-9301190125-1*>
POST: **"chairman emeritus"**
IN_AND_OUT: <*IN_AND_OUT-9301190125-5*>
VACANCY_REASON: OTH_UNK
COMMENT: "Segel in as chmn emeritus at QVC"

<*SUCCESSION_EVENT-9301190125-4*> :=
SUCCESSION_ORG: <*ORGANIZATION-9301190125-2*>
POST: **"chairman"**
IN_AND_OUT: <*IN_AND_OUT-9301190125-6*>
VACANCY_REASON: REASSIGNMENT
COMMENT: "Diller out as chmn at Fox"
/ "This event could be collapsed with *SUCCESSION_EVENT-5*"

<*SUCCESSION_EVENT-9301190125-5*> :=
SUCCESSION_ORG: <*ORGANIZATION-9301190125-2*>

POST: **"chief executive"**
IN_AND_OUT: <*IN_AND_OUT-9301190125-7*>
VACANCY_REASON: REASSIGNMENT
COMMENT: "Diller out as CEO of Fox"

<*SUCCESSION_EVENT-9301190125-6*> :=
SUCCESSION_ORG: <*ORGANIZATION-9301190125-3*>
POST: **"chairman"**
IN_AND_OUT: <*IN_AND_OUT-9301190125-8*>
VACANCY_REASON: REASSIGNMENT
COMMENT: "Diller out as chmn of 20th Century"
/ "This event could be collapsed with *SUCCESSION_EVENT-7*"

<*SUCCESSION_EVENT-9301190125-7*> :=
SUCCESSION_ORG: <*ORGANIZATION-9301190125-3*>
POST: **"chief executive"**
IN_AND_OUT: <*IN_AND_OUT-9301190125-9*>
VACANCY_REASON: REASSIGNMENT
COMMENT: "Diller out as CEO of 20th Century"

<*IN_AND_OUT-9301190125-1*> :=
IO_PERSON: <*PERSON-9301190125-2*>
NEW_STATUS: OUT
ON_THE_JOB: YES
OTHER_ORG: / <*ORGANIZATION-9301190125-1*>
REL_OTHER_ORG: / SAME_ORG
COMMENT: "Segel out as chmn – could be considered to be staying at QVC,
though he apparently won't have any duties there"

<*IN_AND_OUT-9301190125-2*> :=
IO_PERSON: <*PERSON-9301190125-1*>
NEW_STATUS: IN
ON_THE_JOB: NO
OTHER_ORG: <*ORGANIZATION-9301190125-2*>
/ <*ORGANIZATION-9301190125-3*>
REL_OTHER_ORG: OUTSIDE_ORG
COMMENT: "Diller in as chmn – apparently came from
both Fox and 20th Century simultaneously,
(see separate events),
but OTHER_ORG doesn't allow multiple fills"

<*IN_AND_OUT-9301190125-3*> :=
IO_PERSON: <*PERSON-9301190125-2*>
NEW_STATUS: OUT
ON_THE_JOB: YES
OTHER_ORG: / <*ORGANIZATION-9301190125-1*>
REL_OTHER_ORG: / SAME_ORG
COMMENT: "Segel out as CEO"
/ "This object is identical to *IN_AND_OUT-1*"

<*IN_AND_OUT-9301190125-4*> :=

IO_PERSON: <*PERSON-9301190125-1*>
NEW_STATUS: IN
ON_THE_JOB: NO
OTHER_ORG: <*ORGANIZATION-9301190125-2*>
/ <*ORGANIZATION-9301190125-3*>
REL_OTHER_ORG: OUTSIDE_ORG
COMMENT: "Diller in as CEO "
/ "This object is identical to *IN_AND_OUT-2*"

<*IN_AND_OUT-9301190125-5*> :=
IO_PERSON: <*PERSON-9301190125-2*>
NEW_STATUS: IN
ON_THE_JOB: NO
OTHER_ORG: <*ORGANIZATION-9301190125-1*>
REL_OTHER_ORG: SAME_ORG
COMMENT: "Segel in – staying at same org"

<*IN_AND_OUT-9301190125-6*> :=
IO_PERSON: <*PERSON-9301190125-1*>
NEW_STATUS: OUT
ON_THE_JOB: NO
/ UNCLEAR
OTHER_ORG: <*ORGANIZATION-9301190125-1*>
REL_OTHER_ORG: OUTSIDE_ORG
COMMENT: "Diller out as chmn of Fox"

<*IN_AND_OUT-9301190125-7*> :=
IO_PERSON: <*PERSON-9301190125-1*>
NEW_STATUS: OUT
ON_THE_JOB: NO
/ UNCLEAR
OTHER_ORG: <*ORGANIZATION-9301190125-1*>
REL_OTHER_ORG: OUTSIDE_ORG
COMMENT: "Diller out as CEO of Fox"
/ "This object is identical to *IN_AND_OUT-6*"

<*IN_AND_OUT-9301190125-8*> :=
IO_PERSON: <*PERSON-9301190125-1*>
NEW_STATUS: OUT
ON_THE_JOB: NO
/ UNCLEAR
OTHER_ORG: <*ORGANIZATION-9301190125-1*>
REL_OTHER_ORG: OUTSIDE_ORG
COMMENT: "Diller out as chmn of 20th Century"
/ "This object is identical to *IN_AND_OUT-6*"

<*IN_AND_OUT-9301190125-9*> :=
IO_PERSON: <*PERSON-9301190125-1*>
NEW_STATUS: OUT
ON_THE_JOB: NO

/ UNCLEAR
OTHER_ORG: <*ORGANIZATION-9301190125-1*>
REL_OTHER_ORG: OUTSIDE_ORG
COMMENT: "Diller out as CEO of 20th Century"
/ "This object is identical to *IN_AND_OUT-6*"

<*ORGANIZATION-9301190125-1*> :=
ORG_NAME: `"QVC Network Inc."`
ORG_ALIAS: `"QVC"`
ORG_TYPE: COMPANY

<*ORGANIZATION-9301190125-2*> :=
ORG_NAME: `"Fox Inc."`
ORG_TYPE: COMPANY

<*ORGANIZATION-9301190125-3*> :=
ORG_NAME: `"Twentieth Century Fox Film Corp."`
ORG_TYPE: COMPANY

<*PERSON-9301190125-1*> :=
PER_NAME: `"Barry Diller"`
PER_ALIAS: `"Diller"`
PER_TITLE: `"Mr."`

<*PERSON-9301190125-2*> :=
PER_NAME: `"Joseph M. Segel"`
PER_ALIAS: `"Segel"`
PER_TITLE: `"Mr."`

C.6 Sample MUC-6 CO Key

```
<DOC>
<DOCID> wsj93_062.0057 </DOCID>
<DOCNO> 930119-0125.  </DOCNO>
<HL> <COREF ID="1">Diller</COREF> Is Named <COREF ID="0"
TYPE="IDENT" REF="1" STATUS="OPT">Chairman</COREF>,
@ <COREF ID="2" TYPE="IDENT" REF="1" MIN="Executive"
STATUS="OPT">Chief Executive of <COREF ID="4">QVC</COREF></COREF>
</HL>
<DD> 01/19/93 </DD>
<SO> WALL STREET JOURNAL (J), PAGE C25 </SO>
<CO> QVCN </CO>
<IN> LIMITED PRODUCT SPECIALTY RETAILERS (OTS),
ALL SPECIALTY RETAILERS (RTS) </IN>
<DATELINE> WEST CHESTER, Pa. </DATELINE>
<TXT> <p>
<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>.
</p> <p>
<COREF ID="9" TYPE="IDENT" REF="5" MIN="Diller">Mr. Diller, 50 years
old,</COREF> succeeds <COREF ID="24" MIN="Joseph M. Segel">Joseph
M. Segel, who has been named to the post of chairman
emeritus.</COREF> <COREF ID="10" TYPE="IDENT" REF="9"
MIN="Diller">Mr. Diller</COREF> previously was <COREF ID="11"
TYPE="IDENT" REF="10" STATUS="OPT">chairman</COREF> and <COREF
ID="12" TYPE="IDENT" REF="10" MIN="executive" STATUS="OPT">chief
executive of Fox Inc. and Twentieth Century Fox Film Corp., both units
of News Corp.</COREF> <COREF ID="13" TYPE="IDENT"
REF="10">He</COREF> also served for 10 years as <COREF ID="14"
TYPE="IDENT" REF="13" STATUS="OPT">chairman</COREF> and <COREF
ID="15" TYPE="IDENT" REF="13" MIN="executive" STATUS="OPT">chief
executive of Paramount Pictures Corp., a unit of Paramount
Communications Inc.</COREF>
</p> <p>
Arrow Investments Inc., a corporation controlled by <COREF ID="16"
TYPE="IDENT" REF="13" MIN="Diller">Mr. Diller</COREF>, in <COREF
ID="19">December</COREF> agreed to purchase $25 million of <COREF
ID="17" TYPE="IDENT" REF="6">QVC</COREF> stock in a privately
negotiated transaction.  At <COREF ID="18" TYPE="IDENT" REF="19"
MIN="time">that time</COREF>, it was announced that <COREF ID="20"
TYPE="IDENT" REF="16">Diller</COREF> was in talks with <COREF
ID="21" TYPE="IDENT" REF="17">the company</COREF> on becoming
<COREF ID="22" TYPE="IDENT" REF="21">its</COREF> chairman and chief
executive upon <COREF ID="23" TYPE="IDENT" REF="24" MIN="Segel">Mr.
Segel</COREF>'s scheduled retirement this month.

</p> </TXT> </DOC>
```

C.7 Sample MUC-6 CO Response

```
<DOC>
<DOCID> wsj93_062.0057 </DOCID>
<DOCNO> 930119-0125.  </DOCNO>
<HL> Diller Is Named <COREF ID="210000">Chairman</COREF>,
@ <COREF ID="211000" TYPE="IDENT" REF="210000">Chief
Executive</COREF> of QVC </HL>
<DD> 01/19/93 </DD>
<SO> WALL STREET JOURNAL (J), PAGE C25 </SO>
<CO> QVCN </CO>
<IN> LIMITED PRODUCT SPECIALTY RETAILERS (OTS),
ALL SPECIALTY RETAILERS (RTS) </IN>
<DATELINE> WEST CHESTER, Pa.  </DATELINE>
<TXT> <p>
<COREF ID="001">QVC Network Inc</COREF>., as expected, named <COREF
ID="002">Barry Diller</COREF> <COREF ID="210003" TYPE="IDENT"
REF="002"><COREF ID="40003" TYPE="IDENT" REF="001">its</COREF>
chairman</COREF> and <COREF ID="211003" TYPE="IDENT"
REF="210003">chief executive officer</COREF>.
</p> <p>
Mr. <COREF ID="004" TYPE="IDENT" REF="002">Diller, 50 years
old</COREF>, succeeds <COREF ID="005">Joseph M. Segel</COREF>, who
has been named to the post of <COREF ID="006" TYPE="IDENT"
REF="005">chairman</COREF> emeritus.  Mr. <COREF ID="007"
TYPE="IDENT" REF="004">Diller</COREF> previously was <COREF
ID="220008" TYPE="IDENT" REF="007">chairman</COREF> and <COREF
ID="221008" TYPE="IDENT" REF="220008">chief executive of <COREF
ID="40008">Fox Inc.</COREF></COREF> and <COREF
ID="21008">Twentieth Century Fox Film Corp., <COREF ID="151008"
TYPE="IDENT" REF="21008">both units of <COREF ID="51008">News
Corp.</COREF></COREF></COREF> <COREF ID="009" TYPE="IDENT"
REF="10008">He</COREF> also served for 10 years as <COREF
ID="220010" TYPE="IDENT" REF="007">chairman</COREF> and <COREF
ID="221010" TYPE="IDENT" REF="220010">chief executive of <COREF
ID="40010">Paramount Pictures Corp., a unit of <COREF
ID="41010">Paramount Communications</COREF></COREF></COREF>
<COREF ID="21010">Inc.
</p> <p>
Arrow Investments Inc</COREF>., <COREF ID="011">a
corporation</COREF> controlled by Mr. <COREF ID="012" TYPE="IDENT"
REF="007">Diller</COREF>, in December agreed to purchase $25 million
of QVC stock in a privately negotiated transaction.  At that time,
<COREF ID="013" TYPE="IDENT" REF="011">it</COREF> was announced
that <COREF ID="014" TYPE="IDENT" REF="012">Diller</COREF> was in
talks with <COREF ID="015" TYPE="IDENT" REF="013">the
company</COREF> on becoming <COREF ID="210016" TYPE="IDENT"
REF="014"><COREF ID="40016" TYPE="IDENT" REF="015">its</COREF>
chairman</COREF> and <COREF ID="211016" TYPE="IDENT"
REF="210016">chief executive</COREF> upon Mr. <COREF ID="017"
TYPE="IDENT" REF="009">Segel</COREF>'s scheduled retirement this
month.

</p> </TXT> </DOC>
```

ALIAS = *NO*
 SAME-TYPE = *YES*
  PRONOUN-2 = *NO*
   ...
      PRONOUN-1 = *NO*
       NAME-2 = *NO*
        ...
           SAME-SENTENCE = *YES*
            NAME-1 = *YES*

Figure C.1 One branch of RESOLVE's MUC-6 decision tree

IF	*both phrases are the **same type***
AND	*neither phrase is a **pronoun***
AND	*the first phrase includes a **name***
AND	*the second phrase does not include a **name***
AND	*both phrases are in the **same sentence***
THEN	class = **YES** (the phrases are coreferent)

Figure C.2 A rule corresponding to the MUC-6 tree branch in Figure C.1

C.8 Applications of Discovered Knowledge in MUC-6

One branch of the decision tree that RESOLVE learned for the MUC-6 coreference task was shown in Chapter 9, along with a "rule" that that corresponded to a traversal of that tree branch. These figures are reprinted in this section for easy reference (Figures C.1 and C.2).

C.8.1 Applications of the Discovered Rule in the Sample Text

Applications of this learned rule are shown for several sentences from the MUC-6 sample text included in this appendix (Section C.2).

C.8.1.1 Sentence 1

QVC Network Inc., as expected, named Barry Diller
its chairman and chief executive officer.

The discovered rule applied to one instance generated for this sentence, correctly linking the phrase:

its chairman and chief executive officer[1]

to the preceding phrase:

Barry Diller

The pattern represented by this rule application can be described in general terms as:

<*entity*> named <*person-name*> <*person-role*>

C.8.1.2 Sentence 2

Mr. Diller, 50 years old, succeeds Joseph M. Segel, who has been named to the post of chairman emeritus.

The discovered rule applied to one instance generated for this sentence, correctly linking the phrase:

chairman[2]

to the preceding phrase:

Joseph M. Segel

The pattern represented by this rule application can be described in general terms as:

<*person-name*> who has been named to the post of <*person-role*>

[1]This phrase is later split into two separate phrases – **its chairman** and **chief executive officer** – by a postprocessing component that was designed to split complex noun phrases involving appositives and conjunctions into their constituent components, as required by the MUC-6 Coreference Task Definition.

[2]Unfortunately, due to a semantic tagging error, **emeritus** was not recognized as a component of a person's title, and a preprocessing trimmed this word from the phrase, leaving only **chairman**. Since the MUC-6 scoring guidelines gave no partial credit, this would have been marked incorrect.

C.8.1.3 Sentence 3

> Mr. Diller previously was chairman and chief executive of Fox Inc. and Twentieth Century Fox Film Corp., both units of News Corp.

The discovered rule applied to one instance generated for this sentence, correctly linking the phrase:

> chairman and chief executive of Fox Inc. and Twentieth Century Fox Film Corp., both units of News Corp.[3]

to the preceding phrase:

> Diller[4]

The pattern represented by this rule application can be described in general terms as:

<*person-name*> previously was <*person-role*>

C.8.1.4 Sentence 4

> He also served for 10 years as chairman and chief executive of Paramount Pictures Corp., a unit of Paramount Communications Inc.

Unfortunately, the BADGER component for identifying sentence boundaries did not correctly identify this sentence as a separate sentence, since the preceding sentence ended with textstringCorp., a corporate designator abbreviation, i.e., the period at the end of **Corp** was interpreted as part of the abbreviation and not as the end of the sentence.

Fortunately, this resulted in another "correct" application of the discovered rule, linking the phrase:

---

[3] This phrase is later split into two separate phrases – **chairman** and **chief executive** ... – by the postprocessing component that was designed to split complex noun phrases involving appositives and conjunctions into their constituent components.

[4] Due to confusion between the MUC-6 Named Entity Task Definition, which specified that titles such as **Mr.** were to be ***excluded*** in the annotations generated for the system output, and the MUC-6 Coreference Task Definition, which specified that such titles were to be ***included*** in the annotations generated for the system output, the title **Mr.** was trimmed from the phrase **Mr. Diller**. Again, since the MUC-6 scoring software did not give partial credit for substring matches, this would have been marked incorrect.

chairman and chief executive of Paramount Pictures Corp., a unit of Paramount Communications Inc.[5]

to the preceding phrase (from the previous sentence):

Diller[6]

Since the phrases are actually separated by sentence boundaries, this rule application does not represent any pattern that would have been included in the definiteion of the PERSON-IS-ROLE feature. However, if we substitute a proper name reference (**Mr. Diller**) for the pronominal reference (**He**), we might interpret this application as covering the pattern:

<*person-name*> [also] served [for 10 years] as <*person-role*>

C.8.1.5 Sentence 5

Arrow Investments Inc., a corporation controlled by Mr. Diller, in December agreed to purchase $25 million of QVC stock in a privately negotiated transaction.

The discovered rule was not applied to any instances generated in this sentence; there are no examples of intra-sentential coreference in this sentence.

C.8.1.6 Sentence 6

At that time, it was announced that Diller was in talks with the company on becoming its chairman and chief executive upon Mr. Segel's scheduled retirement this month.

The discovered rule applied to one instance generated for this sentence, correctly linking the phrase:

its chairman and chief executive[7]

[5] This phrase is later split into two separate phrases – **chairman** and **chief executive ...** – by the postprocessing component that was designed to split complex noun phrases involving appositives and conjunctions into their constituent components.

[6] Unfortunately, due to the incorrect trimming of **Mr.**, this would have been marked as incorrect by the MUC-6 scoring program.

[7] This phrase is later split into two separate phrases – **its chairman** and **chief executive** – by the postprocessing component that was designed to split complex noun phrases involving appositives and conjunctions into their constituent components.

to the preceding phrase:

```
Diller
```

The pattern represented by this rule application can be described in general terms as:

<*entity*> [was in talks with the company on] becoming <*person-name*> <*person-role*>

C.8.2 Applications of the Discovered Rule in the MUC-6 Walkthrough Text

The entire MUC-6 walkthrough text can be found in the MUC-6 Proceedings [MUC-6, 1995]. The sentences in which the discovered rule was applied will be reprinted here, with a brief narrative for each application.

```
One of the many differences between Robert L. James,
chairman and chief executive officer of McCann-Erickson, and
John J. Dooner Jr., the agency's president and
chief operating officer, is quite telling:  Mr. James enjoys
sailboating, while Mr. Dooner owns a powerboat.
```

The discovered rule compensated for a construct that was incorrectly classified by another MUC-6 system component, the appositive classifier. In this case, the appositive classifier should have linked `John J. Dooner Jr.` with **`the agency's president and chief operating officer`**, and a postprocessing phrase would have split these up into their constituent parts according to the MUC-6 guidelines. Although this appositive construction was missed by the appositive classifier, the discovered rule was able to compensate for this error, correctly linking the two conjoined appositive phrases **`the agency's president`** and **`chief operating officer`** to `John J. Dooner Jr.` The discovered rule thus had two correct applications in this sentence.

```
Now, Mr. James is preparing to sail into the sunset, and
Mr. Dooner is poised to rev up the engines to guide
Interpublic Group's McCann-Erickson into the 21st century.
```

The phrase ***rev***, which was identified as a reference to a person's title by the sentence analyzer (presumably mistaking the word for a shortened form of "Reverend"), was linked to **Mr. Dooner** in this sentence. This incorrect application of the discovered rule was due to the semantic tagging error.

> **Yesterday, *McCann* made *official* what had been widely anticipated: Mr. James, 57 years old, is stepping down *as* chief executive officer on July 1 and will retire as chairman at the end of the year.**

The discovered rule was applied three times in this sentence: linking ***official*** with **McCann**, and linking both **chief executive officer** and **chairman** with **Mr. James, 57 years old**. The first application was incorrect, however, this application was due to a semantic tagging error: **official** was tagged as a generic reference to a person by the sentence analyzer. The other two applications were both correct (although the preposition **as** that precedes **chief executive officer** whould have been trimmed by the preprocessor for RESOLVE).

> **Mr. Dooner, who recently lost 60 pounds over three-and-a-half months, says now that he has *"reinvented"* himself, he wants to do the same for the agency.**

The reflexive pronoun **himself** was correctly linked to **Mr. Dooner**, in another correct application of the learned rule (although the verb ***"reinvented"*** was not properly trimmed in the system response).

> ***McCann* has initiated a new so-called global collaborative system, composed of world-wide account directors paired with creative partners.**

In this sentence, ***McCann*** was incorrectly tagged as a person name – it is the name of an organization in the context of the MUC-6 walkthrough text – which made it the SAME-TYPE as **world-wide account directors** and **creative partners** (although the fact that both of these are *plural* noun phrases was also missed). Both of the latter phrases were incorrectly linked to ***McCann*** in this sentence, but both of these errors are attributable to incorrect semantic tagging by the sentence analyzer.

> In addition, Peter Kim was hired from WPP Group's J. Walter Thompson last September as
> vice chairman, chief strategy officer, world-wide.

The discovered rule correctly linked **vice chairman, chief strategy officer** with **Peter Kim** in this sentence (although the modifier, **world-wide**, was incorrectly trimmed from the annotated output).

> (There are no immediate plans to replace Mr. Dooner as
> president; Mr. James operated as
> chairman, chief executive officer and president for a period
> of time.)

This sentence contains two correct applications of the discovered rule: one linking **president** with **Mr. Dooner** and the other linking **chairman, chief executive officer and president** with **Mr. James**.

> Asked why he would choose to voluntarily exit while he still
> is so ***young***, Mr. James says it is time to be a tad selfish
> about how he spends ***his days***.

The phrase ***young*** was incorrectly identified as a person name by the sentence analyzer, which resulted in the discovered rule being incorrectly in linking the pronoun **his** with that phrase.

> "Coke has given us great highs," says Mr. James, sitting in
> his plush office, filled with photographs of sailing as well
> as ***huge models*** of, among other things, a Dutch tugboat.

The noun ***models*** was incorrectly identified as a person's occupation in this sentence, resulting in the discovered rule incorrectly linking that phrase with **Mr. James**.

BIBLIOGRAPHY

[Aone and Bennett, 1995] Aone, Chinatsu and Bennett, Scott William. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting of the ACL*, 1995.

[Appelt *et al.*, 1992] Appelt, Douglas E., Bear, John, Hobbs, Jerry R., Israel, David, and Tyson, Mabry. SRI International FASTUS system: MUC-4 test results and analysis. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 143–147, 1992.

[Ayuso *et al.*, 1992] Ayuso, Damaris, Boisen, Sean, Fox, Heidi, Gish, Herbert, Ingria, Robert, and Weischedel, Ralph. BBN: Description of the PLUM system as used in MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 169–176, 1992.

[Azzam, 1995] Azzam, Saliha. *Computation of Ambiguities (Anaphors and PPs) in NL texts: CLAM, the Prototype*. PhD thesis, University of Paris, Sorbonne, 1995. (in French).

[Azzam, 1996] Azzam, Saliha. Resolving anaphors in embedded sentences. In *Proceedings of the 34th Annual Meeting of the ACL*, 1996. (to appear).

[Bobrow and Winograd, 1977] Bobrow, Daniel G. and Winograd, Terry. An overview of KRL, a knowledge representation language. *Cognitive Science*, 1(1):3–46, 1977.

[Bogart, 1983] Bogart, Kenneth P. *Introductory Combinatorics*. Pitman Publishing Inc., Marshfield, MA, 1983.

[Boguraev, 1979] Boguraev, B. K. Automatic resolution of linguistic ambuities. TR 11, University of Cambridge Computer Laboratory, 1979.

[Brennan *et al.*, 1987] Brennan, Susan E., Friedman, Marilyn Walker, and Pollard, Carl J. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the ACL*, 1987.

[Brill and Resnik, 1994] Brill, Eric and Resnik, Philip. A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings, COLING-94*, 1994.

[Brill, 1994] Brill, Eric. Some advances in transformation-based part of speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 722–727, 1994.

[Cardie, 1993] Cardie, Claire. Using decision trees to improve case-based learning. In Paul Utgoff, editor, *Proceedings of the Tenth International Conference on Machine Learning*, pages 25–32, University of Massachusetts, Amherst, MA, 1993. Morgan Kaufmann.

[Carter, 1987] Carter, David. *Interpreting Anaphors in Natural Language Texts.* Ellis Horwood Ltd., Chichester, England, 1987.

[Charniak, 1972] Charniak, Eugene. Towards a model of children's story comprehension. AI-TR 266, Artficial Ingelligence Laboratory, Massachusetts Institute of Technology, 1972.

[Chinchor and Sundheim, 1993] Chinchor, Nancy and Sundheim, Beth. MUC-5 evaluation metrics. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 22–29, 1993.

[Chinchor, 1991] Chinchor, Nancy. MUC-3 evaluation metrics. In *Proceedings of the Third Message Understanding Conference (MUC-3)*, pages 17–24, 1991.

[Chinchor, 1992] Chinchor, Nancy. MUC-4 evaluation metrics. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 22–29, 1992.

[Church, 1988] Church, Kenneth W. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143, 1988.

[Cohen, 1995] Cohen, Paul R. *Empirical Methods for Artificial Intelligence.* MIT Press, Cambridge, MA, 1995.

[Collins, 1996] Collins, Michael John. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the ACL*, 1996. (to appear).

[Connolly *et al.*, 1994] Connolly, Dennis, Burger, John D., and Day, David S. A machine learning approach to anaphoric reference. In *Proceedings of the International Conference on New Methods in Language Processing (NEMLAP)*, 1994.

[Fisher *et al.*, 1995] Fisher, David, Soderland, Stephen, McCarthy, Joseph, Feng, Fangfang, and Lehnert, Wendy. Description of the UMass system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, 1995. (to appear).

[Grosz *et al.*, 1983] Grosz, Barbara J., Joshi, Avarind K., and Weinstein, Scott. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting of the ACL*, pages 44–50, 1983.

[Grosz *et al.*, 1986] Grosz, Barbara J., Joshi, Avarind K., and Weinstein, Scott. Towards a computational theory of discourse interpretation. (unpublished manuscript), 1986.

[Grosz *et al.*, 1995] Grosz, Barbara J., Joshi, Avarind K., and Weinstein, Scott. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225, June 1995.

[Grosz, 1977] Grosz, Barbara J. 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, 1977.

[Hirschman, 1992] Hirschman, Lynette. An adjunct test for discourse processing in MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 67–77, 1992.

[Hirst, 1981] Hirst, Graeme. *Anaphora in Natural Language Understanding: A Survey*. Springer-Verlag, Berlin, 1981.

[Hobbs, 1976] Hobbs, Jerry R. Pronoun resolution. Technical report, Department of Computer Science, City College, City University of New York, 1976.

[Hobbs, 1978] Hobbs, Jerry R. Resolving pronoun references. *Lingua*, 44:311–338, 1978.

[Iwańska *et al.*, 1992] Iwańska, Lucja, Appelt, Douglas, Ayuso, Damaris, Dahlgren, Kathy, Glover Stalls, Bonnie, Grishman, Ralph, Krupka, George, Montgomery, Christine, and Riloff, Ellen. Computational aspects of discourse in the context of MUC-3. In *Proceedings of the Third Message Understanding Conference (MUC-3)*, pages 256–282, 1992.

[Krupka and Rau, 1992] Krupka, George and Rau, Lisa. GE adjunct test report: Object-oriented design and scoring for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 78–84, 1992.

[Lehnert *et al.*, 1992] Lehnert, Wendy, Cardie, Claire, Fisher, David, McCarthy, Joe, Riloff, Ellen, and Soderland, Stephen. University of Massachusetts: Description of the CIRCUS system as used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 282–288, 1992.

[Lehnert *et al.*, 1993] Lehnert, W., McCarthy, J., Soderland, S., Riloff, E., Cardie, C., Peterson, J., Feng, F., Dolan, C., and Goldman, S. University of Massachusetts/Hughes: Description of the CIRCUS system as used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 277–290, 1993.

[Litman and Passonneau, 1996] Litman, Diane J. and Passonneau, Rebecca J. Combining multiple knowledge sources for discourse segmentation. In *Proceedings of the 34th Annual Meeting of the ACL*, 1996. (to appear).

[Magerman, 1994] Magerman, David M. *Natural Language Parsing as Statistical Pattern Recognition*. Ph.d. thesis, Stanford University, February 1994.

[Magerman, 1995] Magerman, David M. Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the ACL*. Association for Computational Linguistics, 1995.

[Marcus *et al.*, 1993] Marcus, Mitchell P., Santorini, Beatrice, and Marcinkiewicz, Mary Ann. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[McCarthy and Lehnert, 1995] McCarthy, Joseph F. and Lehnert, Wendy G. Using decision trees for coreference resolution. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1050–1055, Montreal, Canada, August 1995. Morgan Kaufmann.

[MUC-3, 1991] *Proceedings of the Third Message Understanding Conference (MUC-3)*, San Mateo, CA, May 1991. Morgan Kaufmann.

[MUC-4, 1992] *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, San Mateo, CA, June 1992. Morgan Kaufmann.

[MUC-5, 1993] *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, San Mateo, CA, August 1993. Morgan Kaufmann.

[MUC-6, 1995] *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, San Mateo, CA, November 1995. Morgan Kaufmann. (to appear).

[Onyshkevych *et al.*, 1993] Onyshkevych, Boyan, Okurowski, Mary Ellen, and Carlson, Lynn. Tasks, domains, and languages. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 7–17, 1993.

[Ortega and Fisher, 1995] Ortega, Julio and Fisher, Doug. Flexibly exploiting prior knowledge in empirical learning. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1041–1047, 1995.

[Pagallo, 1989] Pagallo, Giulia. Learning DNF by decision trees. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 639–644, 1989.

[Passonneau and Litman, 1993] Passonneau, Rebecca J. and Litman, Diane J. Intention-based segmentation: Human reliability and correlation with linguistic cues. In *Proceedings of the 31th Annual Meeting of the ACL*, pages 148–155, 1993.

[Passonneau, 1994] Passonneau, Rebecca J. Protocol for coding discourse referential noun phrases and their antecedents. Technical report, Columbia University, 1994.

[Passonneau, 1996] Passonneau, Rebecca J. Interaction of the segmental structure of discourse with explicitness of discourse anaphora. In E. Prince, A. Joshi, and M. Walker, editors, *Proceedings of the Workshop on Centering Theory in Naturally Occurring Discourse*. Oxford University Press, 1996. (to appear).

[Pazzani *et al.*, 1994] Pazzani, Michael J., Merz, Christopher, Murphy, Patrick, Ali, Kamal, Hume, Tim, and Brunk, Clifford. Reducing misclassification costs. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 217–225, 1994.

[Quinlan, 1986] Quinlan, J. Ross. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[Quinlan, 1989] Quinlan, J. R. Unknown attribute values in induction. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 164–168, 1989.

[Quinlan, 1990] Quinlan, J. Ross. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[Quinlan, 1993] Quinlan, J. Ross. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[Riloff, 1993] Riloff, Ellen. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 811–816, Washington, DC, July 1993. AAAI Press / MIT Press.

[Sidner, 1979] Sidner, Candace L. Towards a computational theory of definite anaphora comprehension in English discourse. TR 537, M.I.T. Artificial Intelligence Laboratory, 1979.

[Soderland and Lehnert, 1994] Soderland, Stephen and Lehnert, Wendy. Corpus-driven knowledge acquisition for discourse analysis. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, WA, July 1994.

[Sundheim and Dillard, 1987] Sundheim, B. M. and Dillard, R. A. Navy tactical messages: Examples for text-understanding technology. Technical Document 1060, Naval Ocean Systems Center, San Diego, CA, February 1987.

[Sundheim, 1989] Sundheim, Beth. Second message understanding conference (MUCK-II) report. Technical Report 1328, Naval Ocean Systems Center, San Diego, CA, September 1989.

[Sundheim, 1991] Sundheim, Beth M. Overview of the third message understanding evaluation and conference. In *Proceedings of the Third Message Understanding Conference (MUC-3)*, pages 3–16, 1991.

[Sundheim, 1992] Sundheim, Beth M. Overview of the fourth message understanding evaluation and conference. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 3–21, 1992.

[Sundheim, 1993] Sundheim, Beth M. TIPSTER/MUC-5 information extraction system evaluation. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 27–44, 1993.

[Suri, 1993] Suri, Linda Z. Extending focusing frameworks to process complex sentences and to correct the written english of proficient signers of american sign language. TR 94-21, University of Delaware, 1993.

[TIPSTER, 1993] *Proceedings of the TIPSTER Text Program (Phase I)*, San Mateo, CA, September 1993. Morgan Kaufmann.

[van Rijsbergen, 1979] van Rijsbergen, C. J. *Information Retrieval.* Butterworths, London, 1979.

[Walker, 1989] Walker, Marilyn A. Evaluating discourse processing algorithms. In *Proceedings of the 27th Annual Meeting of the ACL*, 1989.

[Weischedel *et al.*, 1993] Weischedel, Ralph, Meteer, Marie, Schwartz, Richard, Ramshaw, Lance, and Palmucci, Jeff. Coping with ambiguity and unknown words through probabilistic models. *Computational Linguistics*, 19(2):359–382, 1993.

[Wilks, 1975] Wilks, Yorick A. A preferential, pattern-seeking semantics for natural language interface. *Artificial Intelligence*, 6:53–74, 1975.

[Will, 1993] Will, Craig A. Comparing human and machine performance for natural language information extraction: Results for English microelectronics from the MUC-5 evaluation. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 53–67, 1993.

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