Knowledge Acquisition for Coreference Resolution

Dissertation
zur Erlangung des akademischen Grades
eines Doktors der Philosophie
der Philosophischen Fakultäten
der Universität des Saarlandes

vorgelegt von Olga Uryupina
aus Moskau

Saarbrücken, 2007

Dekan:	Prof. Dr. Ulrike Demske
Berichterstatter:	Prof. Dr. Manfred Pinkal
	Dr. Mirella Lapata

Tag der letzten Prüfungsleistung: 1.6.2007

Abstract

This thesis addresses the problem of statistical coreference resolution. Theoretical studies describe coreference as a complex linguistic phenomenon, affected by various different factors. State-of-the-art statistical approaches, on the contrary, rely on rather simple knowledge-poor modeling. This thesis aims at bridging the gap between the theory and the practice.

We use insights from linguistic theory to identify relevant linguistic parameters of co-referring descriptions. We consider different types of information, from the most shallow name-matching measures to deeper syntactic, semantic, and discourse knowledge. We empirically assess the validity of the investigated theoretic predictions for the corpus data. Our data-driven evaluation experiments confirm that various linguistic parameters, suggested by theoretical studies, interact with coreference and may therefore provide valuable information for resolution systems. At the same time, our study raises several issues concerning the coverage of theoretic claims. It thus brings feedback to linguistic theory.

We use the investigated knowledge sources to build a linguistically informed statistical coreference resolution engine. This framework allows us to combine the flexibility and robustness of a machine learning-based approach with wide variety of data from different levels of linguistic description.

Our evaluation experiments with different machine learners show that our linguistically informed model, on the one side, outperforms algorithms, based on a single knowledge source and, on the other side, yields the best result on the MUC-7 data, reported in the literature (F-score of 65.4% with the SVM^{light} learning algorithm).

The learning curves for our classifiers show no signs of convergence. This suggests that our approach makes a good basis for further experimentation: one can obtain even better results by annotating more material or by using the existing data more intelligently.

Our study proves that statistical approaches to the coreference resolution task may and should benefit from linguistic theories: even imperfect knowledge, extracted from raw text data with off-the-shelf error-prone NLP modules, helps achieve significant improvements.

Zusammenfassung

Diese Arbeit befasst sich mit dem Problem der statistischen Koreferenzauflösung. Theoretische Studien bezeichnen Koreferenz als ein vielseitiges linguistisches Phänomen, das von verschiedenen Faktoren beeinflusst wird. Moderne statistiche Algorithmen dagegen basieren sich typischerweise auf einfache wissensarme Modelle. Ziel dieser Arbeit ist das Schließen der Lücke zwischen Theorie und Praxis.

Ausgehend von den Erkentnissen der theoretischen Studien erfolgt die Bestimmung der linguistischen Faktoren die fuer die Koreferenz besonders relevant erscheinen. Unterschiedliche Informationsquellen werden betrachtet: von der Oberflächenübereinstimmung bis zu den tieferen syntaktischen, semantischen und pragmatischen Merkmalen. Die Präzision der untersuchten Faktoren wird mit korpus-basierten Methoden evaluiert. Die Ergebnisse beweisen, dass die Koreferenz mit den linguistischen, in den theoretischen Studien eingebrachten Merkmalen interagiert. Die Arbeit zeigt aber auch, dass die Abdeckung der untersuchten theoretischen Aussagen verbessert werden kann.

Die Merkmale stellen die Grundlage für den Aufbau eines einerseits linguistisch gesehen reichen andererseits auf dem Machinellen Lerner basierten, d.h. eines flexiblen und robusten Systems zur Koreferenzauflösung.

Die aufgestellten Untersuchungen weisen darauf hin dass das wissensreiche Model erfolgversprechende Leistung zeigt und im Vergleich mit den Algorithmen, die sich auf eine einzelne Informationsquelle verlassen, sowie mit anderen existierenden Anwendungen herausragt. Das System erreicht einen F-wert von 65.4% auf dem MUC-7 Korpus. In den bereits veröffentlichen Studien ist kein besseres Ergebnis verzeichnet.

Die Lernkurven zeigen keine Konvergenzzeichen. Somit kann der Ansatz eine gute Basis fuer weitere Experimente bilden: eine noch bessere Leistung kann dadurch erreicht werden, dass man entweder mehr Texte annotiert oder die bereits existierende Daten effizienter einsetzt.

Diese Arbeit beweist, dass statistiche Algorithmen fuer Koreferenzauflösung stark von den theoretischen linguistischen Studien profitiern können und sollen: auch unvollständige Informationen, die automatische fehleranfällige Sprachmodule liefern, können die Leistung der Anwendung signifikant verbessern.

Acknowledgments

Thanking all those who helped me during all these years of writing this thesis is a hard but very pleasant duty.

I am very grateful to my supervisors, Manfred Pinkal and Mirella Lapata, for their support, advice and patience. Their comments have largely improved the results and the presentation. Manfred has given helpful advice on the theoretical studies related to my work. He has also tried to teach me think positively, which was a really challenging task. Mirella's criticism and very detailed comments have made this thesis much better. Mirella and Frank Keller have given me invaluable support on the statistical part of the thesis.

I thank Ewan Klein for his comments on the name-matching part and Massimo Poesio for our discussions on discourse-new entities. Special thanks to James Curran and Stephen Clark for re-training their NE-tagger on the MUC data.

I have been lucky to be a member of the International Graduate College. This thesis has benefited a lot from our seminars, annual meetings and after-hours discussions. Many thanks to Matt Crocker and other IGK professors for creating such a great working environment and to all the fellow IGK students both in Saarbrücken and in Edinburgh.

Finally, I would like to thank my family and my friends, who helped me during these years: Markus, Carsten, Olga, Yurii, Suren, Peter, Amit, Malte and Ute, Annabel, Alexander, Vladimir, Dominika, Rebecca, Kristina, and Sandra. Special thanks to Vova for supporting me at the final steps of the marathon.

Contents

1 Introduction **1**
- 1.1 Contributions . . . 6
- 1.2 Overview of the Thesis . . . 8

2 Methodology **11**
- 2.1 Learning-based Coreference Resolution with a Rich Feature Set 12
- 2.2 Data . . . 15
- 2.3 Baseline . . . 26
- 2.4 External Modules . . . 29
- 2.5 Markables . . . 32
- 2.6 Summary . . . 34

3 Surface Similarity: Name Matching **35**
- 3.1 Related Work . . . 37
- 3.2 Challenges for Name-Matching Algorithms . . . 38
- 3.3 Computing Surface Similarity . . . 41
- 3.4 Features and Their Configurations . . . 46
- 3.5 Experiments . . . 52
  - 3.5.1 Experiment 1: Instance-based Evaluation . . . 53
  - 3.5.2 Experiment 2: MUC-style Set-based Evaluation . . . 54
  - 3.5.3 Discussion . . . 54
- 3.6 Summary . . . 58

4 Syntactic Knowledge **59**
- 4.1 Related Work . . . 60
- 4.2 Types of Markables . . . 62
- 4.3 Determiner . . . 63
- 4.4 Head . . . 66
- 4.5 Internal Structure of Markables . . . 70

4.6 Intrasentential Constraints . . . 72
4.7 Explicit Indicators for Coreference . . . 77
4.8 Grammatical Roles . . . 83
4.9 Morphological Agreement . . . 87
4.10 Experiments . . . 89
4.10.1 Experiment 3: Syntactic Knowledge for Intrasentential Anaphora Resolution . . . 90
4.10.2 Experiment 4: Syntactic Knowledge for Full-Scale Coreference Resolution . . . 94
4.11 Summary . . . 95

5 Semantic Compatibility **97**
5.1 Related Work . . . 99
5.2 Semantic Class and Gender Agreement . . . 101
5.3 Semantic Similarity . . . 109
5.4 WordNet Configurations . . . 112
5.5 Experiments . . . 114
5.5.1 Experiment 5: Using Semantic Knowledge to resolve common NPs . . . 117
5.5.2 Experiment 6: Semantics-based Full-scale NP-coreference Resolution . . . 121
5.6 Summary . . . 123

6 Discourse Structure and Salience **125**
6.1 Related Work . . . 126
6.2 Document Structure . . . 128
6.3 Discourse Levels: Embedded Sub-discourses . . . 133
6.4 Proximity . . . 138
6.5 Centering-based Discourse Properties . . . 142
6.6 Salient Candidates . . . 145
6.7 Coreferential Status of Candidate Antecedents . . . 146
6.8 Experiments . . . 153
6.8.1 Experiment 7: Salience-based Pronoun Resolution . . . 154
6.8.2 Experiment 8: Salience-based NP-Coreference Resolution 165
6.9 Summary . . . 167

7 Anaphoricity and Antecedenthood **169**
7.1 Related Work . . . 170
7.2 Experiments . . . 174
7.2.1 Experiment 9: Identifying Non-anaphors and Non-antecedents . . . 174
7.2.2 Experiment 10: Integrating Anaphoricity and Antecedenthood Prefiltering into a Coreference Resolution Engine 183

7.3 Summary 185

8 Combining Different Knowledge Types 187
8.1 Experiment 11: Coreference Resolution with and without linguistic knowledge 188
8.1.1 Baselines 189
8.1.2 Performance and Learning Curves 192
8.2 Error Analysis 194
8.2.1 Recall Errors 195
8.2.2 Precision Errors 219
8.3 Discussion 235
8.4 Summary 239

9 Conclusion 241
9.1 Main Findings 241
9.2 Future Work 243

A List of Features 247

Bibliography 257

Chapter 1

Introduction

When people create a text, they want to convey information. It is essential for the speaker to organize her text in such a way, that the hearer is able to reconstruct the message and relate it to his background knowledge. Documents, created by human writers for human readers[1], are therefore not arbitrary collections of sentences, but units exposing specific structural properties.

Theoretical studies on discourse structure have identified various linguistic properties that distinguish a coherent text from a random sequence of sentences (for example, (Halliday and Hasan, 1976), see Section 6.1 for an overview). In particular, a coherent document concentrates on a few central "entities":

(1) One reason Lockheed Martin Corp. did not announce a full acquisition of Loral Corp. on Monday, according to Bernard Schwartz, Loral's chairman, was that Lockheed could not meet the price he had placed on Loral's 31 percent ownership of Globalstar Telecommunications Ltd. Globalstar plans to provide telephone service by bouncing signals off 48 low-orbiting satellites. But with no customers expected until 1998, the need for nearly $2 billion in investment and numerous competitors lurking in the shadows, Globalstar's prospects would not appear to be valuable to the average Lockheed shareholder. Still, Schwartz feels differently, and so now do many investors.

This snippet is organized around a person, "Bernard Schwartz", and a bunch of companies – "Lockheed Martin", "Loral", and "Globalstar". All these names

[1] A document, created for other purposes, may occasionally be very fragmentary. For example, "doorway" web pages, used for search engine optimization and aimed at web crawlers, often contain completely unrelated sentences.

are repeated several times throughout the text, determining its topic. A document, only mentioning new entities, sounds fragmentary and is hard to follow:

(2) One reason Lockheed Martin Corp. did not announce a full acquisition of Loral Corp. on Monday, according to Bernard Schwartz, Boeing's chairman, was that Aerospatiale could not meet the price Louis Gallois had placed on Vought systems' 31 percent ownership of Globalstar Telecommunications Ltd.
Eurocopter plans to provide telephone service by bouncing signals off 48 low-orbiting satellites. But with no customers expected until 1998, the need for nearly $2 billion in investment and numerous competitors lurking in the shadows, McDonnell Douglas Corp.'s prospects would not appear to be valuable to the average Hughes shareholder. Still, Virnell Bruce feels differently, and so now do many investors.

We have created this snippet by deliberately replacing all the subsequent mentions of the central entities in (1) with another names from the same domain. It is definitely much more difficult to understand the message behind (2) than the one behind (1). A clearly identifiable topic (or small set of topics) is essential for establishing global coherence.

A discourse entity is normally not referred to with the identical expression throughout a document:

(3) One reason Lockheed Martin Corp. did not announce a full acquisition of Loral Corp. on Monday, according to Bernard Schwartz, Loral Corp.'s chairman, was that Lockheed Martin Corp. could not meet the price Bernard Schwartz had placed on Loral Corp.'s 31 percent ownership of Globalstar Telecommunications Ltd.
Globalstar Telecommunications Ltd. plans to provide telephone service by bouncing signals off 48 low-orbiting satellites. But with no customers expected until 1998, the need for nearly $2 billion in investment and numerous competitors lurking in the shadows, Globalstar Telecommunications Ltd.'s prospects would not appear to be valuable to the average Lockheed Martin Corp. shareholder. Still, Bernard Schwartz feels differently, and so now do many investors.

We have created this example by editing all the mentions of the central entities to have exactly the same surface form. This text, again, is much more difficult to read than (1). We should have used more pronouns and simplified descriptions to explicitly establish dependencies between sentences. Correctly selected surface representations of discourse entities are essential for establishing local coherence.

The closely related, but still distinct notions of coreference and anaphora

have been advocated by theoretical studies on discourse structure to account for the phenomena, illustrated in our Examples (1–3). Two descriptions M_i and M_j are *coreferential* if they denote the same object. There is no agreement among linguists on the exact definition of *anaphora*. For example, Webber (1979) considers a description M_i to be an *anaphor*, and M_j – its *antecedent*, if the interpretation of M_i depends in some way on M_j. Kamp and Reyle (1993) suggest a similar broad view but then restrict the scope of anaphora to only cover the descriptions, coreferential with their antecedents, i.e. the relation between an anaphor and its antecedent can only be the identity of reference.

Most application-oriented approaches to anaphora resolution follow Kamp and Reyle (1993) and account exclusively for different kinds of (pro-)nominal anaphors, coreferential with their antecedents. It makes anaphora (in this narrow sense) a sub-problem of coreference. For example, "he" in (1) is an anaphor, having "Bernard Schwartz" as its antecedent, and, at the same time, "he" and "Bernard Schwartz" are coreferential.

This study is devoted to coreference. We will however use the terms "anaphor" and "antecedent" throughout this thesis, to denote the second (following) and the first (preceding) nominal description in a coreference link, even if the second mention can be interpreted independently on the first. For example, Loral Corporation is mentioned three times in (1): the first description ("Loral Corp.") is an antecedent, the third description ("Loral" in "Loral's 31 percent ownership") is an anaphor, and the second description ("Loral" in "Loral's interest") is an anaphor for the first one and an antecedent for the third one at the same time. The term "anaphora resolution" will be used synonymous with "coreference resolution" in the following chapters.

We discuss the notions of coreference and anaphora and the corresponding natural language processing tasks in detail below.

Coreference resolution. Our thesis advocates a corpus-based analysis. We have therefore to rely on the definition of coreference adopted for annotating our corpus data (MUC-7). The guidelines of the Message Understanding Conference (Hirschman and Chinchor, 1997) define coreference as a relation on pairs of nominal descriptions in a document. Two descriptions are coreferent if, first, they are both referential and, second, they refer to the same discourse entity[2].

The first condition rules out *non-referring* noun phrases (Karttunen, 1976), such as "no customers" in (1). The second condition induces a partition of all the referring markables into coreference classes, or "chains", corresponding to discourse entities. Coreference is, by definition, an equivalence relation.

The task of coreference resolution consists in identifying all the chains in

[2]See Section 2.2 for more details on the MUC-7 annotation guidelines.

a document. Coreference resolution systems are potentially useful for a variety of natural language processing tasks. For example, they provide important data for Information Extraction engines: a coreference resolution module helps us collect mentions of a given entity throughout the document and thus merge the relevant knowledge, extracted from different text parts. The importance of coreference resolution modules for Information Extraction systems has motivated the decision to establish a special coreference track at the MUC-6 and MUC-7 IE competitions.

Coreference resolution is important for multi-document coreference and information retrieval engines. These systems collect and compare data from different sources. For example, they may extract all the documents in a large corpus, mentioning "Bernard Schwartz". It is essential for such approaches to employ a prediction function that could tell whether "Bernard Schwartz" in Document 1 and "Bernard Schwartz" in Document 2 are indeed the same person. It can be shown (Bagga and Baldwin, 1998b) that such functions can be designed significantly more accurately if one takes into account all the mentions of the entity, i.e. intra-document coreference chains.

Anaphora resolution. Webber (1979) and Kamp and Reyle (1993) explain the notion of anaphora as follows. The interpretation of each new sentence in a text must rely on two kinds of structures, the syntactic structure of the sentence itself and the structure representing the context of the earlier sentences. Elements of the sentence, that rely on the latter structure for their interpretation, are called *anaphoric*. Kamp and Reyle (1993) mainly focus on pronominal anaphors, coreferential with their antecedents, highlighting however a few other types of anaphoric descriptions (including, for example, definites and temporal expressions).

We cannot reconstruct the referent of an anaphoric description taken out of its context. For example, the pronoun "he" in (1) can refer to any (real or imaginary) man. It is the context that disambiguates "he" to "Bernard Schwartz". Note that the original definition only assumes that anaphoric expressions are somehow related to their context. In particular, anaphors, in a broad sense, do not have to be coreferential with their antecedents:

(4) A. [A man] and a woman entered the Golden Eagle. [The man] was wearing a brown overcoat.
B. [My car] isn't running. [The carburettor] malfunctions.

Example (4A) illustrates the most common situation: an anaphoric description, "the man", corefers with its antecedents, "A man". The vast majority of research on anaphora only accounts for such cases. Example (4B), however, shows that a clearly anaphoric description, "the carburettor", may be connected to its antecedent by a different relation (so-called "bridging" or

"associative anaphora" in this case).

The task of anaphora resolution, as understood by Webber (1979) and others, consists in identifying anaphoric expressions, interpreting contextual clues to find their antecedents and classifying the relations between the pairs. It has a much larger scope than the coreference resolution task described before. Moreover, different anaphors may rely on very different clues and raise different problems. State-of-the-art systems, therefore, focus on specific types of anaphors: for example, pronouns (see an overview of Mitkov (1999)), definite noun phrases (Vieira and Poesio, 2000), other-NPs (Modjeska et al., 2003) or descriptions with clause-level antecedents (Webber, 1979; Webber et al., 2003).

Anaphoric links between coreferential descriptions (4A), investigated, for example, by Kamp and Reyle (1993), can be established by a full-scale coreference resolution systems. Other types of anaphoric links are essentially more difficult. Poesio and Vieira (1998) have shown, for example, that even human readers strongly disagree on interpreting "bridging anaphora" (4B). In other words, the task of anaphora resolution can be split into two parts: a sub-task of full-scale coreference resolution and a much more difficult and less clearly definable problem of tackling non-coreferential anaphoric links. Even the former, however, can be beneficial for a variety of language processing algorithms.

Anaphora resolution systems are crucial for text understanding. They provide information that helps us disambiguate semantically vague descriptions, in particular, pronouns. Such knowledge is necessary for deep analysis of a document, for example, for building sophisticated discourse models.

Studies on anaphora resolution bring us valuable knowledge on how naturally occurring texts are (locally) organized. This information can be used for generation purposes. We can significantly improve the readability of an automatically generated document by choosing correct simplified descriptions (recall our example (3)). This involves modeling anaphora resolution for the hearer (see, for example, (Striegnitz, 2004)).

Anaphora resolution can be useful for shallow applications as well, for example, for machine translation. Pronominal anaphors typically agree in gender with their antecedents:

(5) A. ENGLISH: Once upon a time I wrote [a thesis]. You are reading [it] now.
GERMAN: Es war einmal als Ich [eine Doktorarbeit] schrieb. Sie lesen [die] jetzt.
B. ENGLISH: Once upon a time I wrote [a book]. You are reading [it] now.
GERMAN: Es war einmal als Ich [ein Buch] schrieb. Sie lesen [es] jetzt.
C. ENGLISH: Once upon a time I wrote [a novel]. You are reading [it] now.
GERMAN: Es war einmal als Ich [einen Roman] schrieb. Sie lesen [den]

jetzt.

The same English pronoun "it" is translated with different German pronouns "die", "es", and "den", depending on the gender of its antecedent. One can augment a shallow machine translation engine with a pronoun resolution module to help it pick correct gender forms for pronouns in the target language. Different languages may, in addition, have different pronominalization strategies. We might therefore want to translate a pronoun in our source language with a nominal description in our target language. We have therefore to employ anaphora resolution/generation modules for both languages if we want to produce coherent documents.

Text summarization engines may also benefit from anaphora resolution. These systems extract relevant snippets from different parts of one or several documents and arrange them in a single "summary". Extracted snippets should not, obviously, contain anaphors without their antecedents. For example, the final sentence of (1), "Still, Schwartz feels differently, and so now do many investors." cannot be correctly understood without its context. A text summarization system should at least be able to identify anaphoric expressions ("Still", "differently", and "so" in our example[3]). The next step would consist in resolving the anaphors to concentrate only on those, that do not have antecedents within the selected element ("Still" and "differently"). A shallow summarization engine can simply discard elements containing such descriptions. A more advanced system can try to eliminate some anaphors ("Still") or replace them with there antecedents ("differently"). It is a challenging problem, requiring both complex anaphora resolution and text rewriting strategies. Barzilay and Lapata (2005) have shown, however, that even very simple techniques for anaphora and coreference resolution may help model coherence for automatically created summaries.

To summarize, coreference and anaphora are two distinct, though related phenomena. In this study we focus on coreference resolution — the task of partitioning nominal descriptions in a document into equivalence classes, corresponding to discourse entities.

1.1 Contributions

Theoretical studies identify numerous linguistic factors relevant for coreference and anaphora resolution[4]. State-of-the-art statistical approaches to the task,

[3] "Schwartz" would not be considered an anaphor by most existing studies. It is however clear that its meaning is, at least in some sense, dependent on the full form, "Bernard Schwartz", and a good summary should start with the full name.

[4] Chapters 3, 4, 5, and 6, dedicated to different knowledge types, start with an overview of relevant theoretical studies.

on the contrary, rely on very few (10-20) simple features. In this thesis we want to bridge the gap between the theory and the practice, by incorporating sophisticated linguistic knowledge into a data-driven coreference resolution engine. We evaluate our algorithm for a variety of machine learners on a standard dataset (MUC-7) with a traditional learning set-up (Soon et al., 2001) to investigate the usability of linguistically motivated features.

Coreference resolution task is an important research topic, widely addressed in the literature in the past two decades, but the existing algorithms still only have a moderate performance (around 60% F-measure for coreference chains on the MUC-7 data). Cristea et al. (2002) claim that the main problem comes from "tricky anaphors" that state-of-the-art coreference resolution systems still cannot successfully handle – most systems successfully resolve essentially the same set of "easy links". We see two possible solutions to the problem: one can either choose a more sophisticated statistical model or give better knowledge (more elaborated features) to the system.

Various recent studies have investigated the first possibility — extending or significantly changing the resolution strategy: for example, advocating sample selection (Harabagiu et al., 2001; Ng and Cardie, 2002a; Uryupina, 2004), clustering (Cardie and Wagstaff, 1999), Bell trees (Luo et al., 2004), or sequence modeling with Conditional Random Fields (McCallum and Wellner, 2003). The second possibility, giving the algorithm more knowledge, has not been investigated in a sufficient way so far. A remarkable exception is the approach with 53 features advocated by Ng and Cardie (2002c): the authors report improvement, however, only with manual feature selection and after adjusting the modeling scheme.

The goal of the present thesis is to investigate the usability of linguistic factors, suggested by theoretical studies, for automatic coreference resolution. We list the main contributions of the thesis below.

First, we have conducted an extensive corpus-based evaluation of linguistic parameters, suggested by numerous theoretical studies on coreference and anaphora. We have computed distributions for name-matching, syntactic, semantic, and salience properties of NPs and NP pairs and investigated their interaction with coreference. We have identified a number of problems occurring when theoretical predictions are applied to real-world data. These findings may provide valuable feedback for theoretical studies, leading to more accurate claims on linguistic cues for coreference.

Second, we have proposed a robust and scalable learning-based coreference resolution algorithm, incorporating different kinds of linguistic knowledge. It relies on 351 nominal feature (1096 boolean/continuous), representing surface, syntactic, semantic and salience-based properties of markables and markables' pairs. All the values are computed fully automatically. Our evaluation experiments show that a linguistically informed model brings significant improvements over simpler algorithms, provided the underlying machine learning mod-

ule has built-in control for over-fitting.

Third, we have implemented our algorithm in a full-scale coreference resolution engine and evaluated it on a standard dataset (MUC-7). To our knowledge, this system achieves the best performance level for the MUC-7 corpus, reported in the literature (F-score of 65.4%). The learning curves show no signs of convergence. This makes us believe that even better performance figures can be achieved by annotating more material. This also makes our system a good starting point for further machine learning experiments.

We have also developed a linguistically-motivated learning-based algorithm for anaphoricity and antecedenthood detection, showing a reliable performance level on the MUC data (80% F-score for non-antecedents and 88% for non-anaphors on the test corpus).

Finally, we have performed an extensive error analysis. To our knowledge, most studies on coreference do not attempt any systematic and detailed investigation of the errors made by their systems. Our analysis identifies problematic areas and raises numerous issues for future research. It is therefore helpful for application-oriented approaches to coreference.

To summarize, the main goal of this thesis is to investigate the applicability of numerous theoretical claims for large-scale coreference resolution. Our study presents a robust model of coreference, at the same time incorporating complex linguistic factors. It shows that knowledge from different linguistic theories can contribute to data-driven coreference resolution and, vice versa, corpus-based analysis provides valuable feedback for theoretical studies on coreference. Our evaluation experiments confirm that a linguistically informed model can outperform its knowledge-poor counterparts. Our rich feature set has allowed us to create a coreference resolution system yielding the best performance figures reported so far in the literature for the MUC-7 data. The algorithm can be improved further by shifting to more complex processing strategies.

1.2 Overview of the Thesis

This thesis has three parts. Chapter 2 addresses methodological issues. In Chapters 3–6 we investigate different knowledge types and their role for coreference resolution. Chapters 7 and 8 are devoted to combining the investigated information to build statistical models for anaphoricity/antecedenthood detection and for full-scale coreference resolution.

Chapter 2 outlines the methodology of our study. We explain the motivation behind our learning-based approach relying on a rich linguistically motivated feature set. We describe the MUC-7 corpus, used in all the experiments throughout the thesis. We also give an overview of language processing tools and lexical resources that help us extract linguistic knowledge from the raw text data. Finally, we introduce our algorithm for extracting "markables"

— basic units, on which any coreference resolution system operates.

Chapter 3 focuses on the most shallow information — the surface form of markables. We analyze typical name-matching problems and decompose the task of comparing the surface strings, representing our markables, into three steps: normalization, substring selection, and matching proper. We combine different solutions to these sub-problems to come up with a relatively large set of matching features. Our evaluation experiments show that this more sophisticated matching algorithm significantly outperforms simpler surface and head matching strategies, adopted by state-of-the-art coreference resolution systems. It is also reliable enough to serve as a backbone for a full-scale coreference resolution engine.

Chapter 4 evaluates the usability of syntactic knowledge for coreference resolution. We investigate the distributions of markables' types, determiners, heads, modification patterns and grammatical roles and their interaction with coreference. We also investigate the predictive power of several indicators for and against coreference: appositions and copulas, syntactic parallelism, number/person agreement and command relations. We run two learning experiments to assess the importance of syntactic knowledge for intrasentential anaphora and for full-scale coreference resolution. The experiments show that syntactic features are reliable, though seldom applicable, predictors for coreference.

Chapter 5 investigates semantic properties of coreferring descriptions. We evaluate three ways of modeling semantic compatibility: semantic class (including gender) agreement, WordNet similarity, and specific data-driven patterns of WordNet subgraphs. We run two learning experiments to assess the importance of semantic knowledge for nominal anaphora and for full-scale coreference resolution. The experiments suggest that WordNet-based coreference classifiers yield low precision by considering too many descriptions semantically compatible. We can, however, slightly improve the performance of the baseline system by augmenting it with semantic features.

Chapter 6 exploits discourse and salience properties of coreferring descriptions. We investigate the interaction of document structure and proximity with coreference. We discuss several basic criteria for defining salient descriptions. We then turn to more complex discourse properties, advocated by the centering theory. We finally shift from our markable-level model of salience to a more advanced entity-level model. Two experiments assess the importance of salience features for pronominal anaphora and for full-scale coreference resolution. We show that a salience-based approach significantly outperforms the baseline for pronominal anaphora, but is a poor predictor for the full-scale coreference resolution task. This is in accordance with the theoretical claims that different types of anaphors rely on different contextual clues and, therefore, the task of full scale coreference resolution cannot be achieved by relying on a single knowledge source.

Chapter 7 is devoted to anaphoricity and antecedenthood modeling. The tasks consist in detecting likely anaphors and antecedents prior to the actual resolution. We show that a linguistically informed model, combining surface, syntactic, semantic and salience features with the parameters suggested by Karttunen (1976), provides a reliable solution for the both tasks. Similar performance, however, can be achieved with simpler methods, relying on a combination of syntactic features and surface matching.

Chapter 8 describes our linguistically informed coreference resolution engine. We combine the investigated name-matching, syntactic, semantic, and discourse properties in a rich feature set and train a family of classifiers to assess the influence of linguistic knowledge on a statistical coreference model. Our evaluation experiments show that a linguistically-informed algorithm outperforms, first, single-source classifiers (created for each knowledge group separately), and, second, yields the best results on the MUC-7 data reported so far in the literature. We also present a detailed error analysis, raising several issues , mostly not addressed in the literature.

Chapter 9 outlines the main findings of the thesis and suggests possible directions for future work.

Chapter 2

Methodology

The task of coreference resolution can be addressed from different perspectives. Rule-base approaches (Wilks 1973; Carter 1987; Alshawi 1992, among others) focus on relevant linguistic information. This involves constructing sophisticated discourse models that can potentially account for very complex cases of anaphora:

(6) Bugs and Bunny are rabbits. $[\text{Bugs}]_1$ is in the hat. John removes $[\text{the birthday present}]_{2,ante=1}$ from the hat.

Gardent and Konrad (1999) present a model-theoretic approach to coreference: their system relies on a deep semantic representation of (6) to infer that "the birthday present" and "Bugs" both describe the only object "in the hat" and, therefore, are coreferent. Such algorithms rely on sophisticated inference schemes with large hand-coded knowledge bases. It is a very time-consuming task to create such a system for a new domain and therefore rule-based algorithms can hardly serve as a backbone for general-purpose coreference resolution engines, limiting the scalability of rule-based approaches.

Data-driven algorithms (Cardie and Wagstaff 1999; Soon et al. 2001; Strube et al. 2002a; McCallum and Wellner 2003; Luo et al. 2004, among others), on the contrary, rely on very few shallow linguistic parameters. Such studies concentrate mostly on improving resolution strategies – investigating different sampling techniques and searching for the most suitable machine learning algorithm. Data-driven approaches, unlike their rule-based counterparts, can be easily adjusted (re-trained) to cover new datasets, but they only account for very easy cases of anaphora – no sophisticated learning scheme can compensate for the lack of relevant knowledge.

This thesis aims at combining the main advantages of the two trends. We present a data-driven algorithm relying on a rich pool of features. We believe that such a setting allows us to create a coreference resolution system that, on the one hand, is robust and scalable, but, on the other hand, can account for difficult anaphoric links.

Our goal is, first, to investigate the interaction between various linguistic parameters of noun phrases and NP pairs and, second, to combine the relevant parameters, encoded as features, and build a linguistically-motivated coreference resolution engine.

Below we discuss the methodology used throughout this thesis in details and introduce external resources we have relied upon.

2.1 Learning-based Coreference Resolution with a Rich Feature Set

This thesis advocates a corpus-based approach to the coreference resolution task. We investigate the interaction of various linguistic parameters with coreference, encode relevant parameters as features and build several learning-based anaphora resolution systems to assess the importance of different kinds of linguistic knowledge.

Our approach relies on two key points: it is a *learning-based* algorithm with a *rich feature set*. It differs, on the one hand, from rule-based approaches (Carter 1987; Wilks 1973; Alshawi 1992, among others) and, on the other hand, from knowledge-poor systems (Kennedy and Boguraev 1996; Mitkov 1998; Cardie and Wagstaff 1999, among others).

Rule-based vs. Learning-based Coreference Resolution. Most existing coreference resolutions algorithms[1] follow the same two-step processing scheme: they start by classifying *pairs* of noun phrases in a document as [±coreferent] and then intelligently *merge* these decisions to construct coreference chains (see Section 2.3 for an example of such approach). Rule-based and learning-based algorithms differ at their first step: the former rely on a set of hand-crafted heuristics and the latter employ a prediction function acquired automatically by some machine learning algorithm.

We see several reasons to opt for a learning-based approach. First, we do not only aim at building a reliable coreference resolution system, but also want to understand the role of different knowledge types. This goal can hardly be achieved within a rule-based framework: the same knowledge can be used by several heuristics and therefore one might need to significantly readjust the system in order to assess the impact of a particular information source.

[1]The only exceptions among the systems mentioned throughout the thesis are the algorithms proposed by Cardie and Wagstaff (1999) and Luo et al. (2004).

Learning-based systems, on the contrary, allow to explicitly compare the contribution of different information sources by manipulating their feature sets.

Second, learning-based algorithms are more flexible — they can be retrained for a new corpus or domain with only minimal adjustments. Porting a rule-based system to cover a new dataset is a very notorious task, especially when the data contain rare or even erroneous patterns. Creating a rule-based system for a large corpus involves a lot of time-consuming manual knowledge engineering and is therefore hardly feasible.

Third, learning-based approaches are more robust, as we train our classifiers on real-world texts. Rule-based systems, on the contrary, are typically created by examining sets of pre-selected or even manually crafted examples and therefore may have low coverage.

Finally, empirical evaluation experiments show that data-driven algorithms for coreference resolution outperform their rule-based counterparts. This could be partially explained by the latest trends in the coreference resolution community: rule-based systems are generally older and therefore may not reflect the most recent findings in the field. Even the first empiric studies, however, suggest a clear preference for learning-based processing. For example, McCarthy and Lehnert (1995) report that their statistical algorithm outperforms their rule-based system by around 8% on the MUC-5 data.

We should also keep in mind possible disadvantages of statistical approaches — coreference is intrinsically difficult for machine learning. First, the distribution of [±coreferent] pairs is highly skewed: if we include *all* the possible {NP_i, NP_j} pairs into the training set, it will contain around 99% of non-coreferent instances. Sample selection techniques may help us to reduce the amount of negative instances, but the resulting distribution is still too biased towards the [−coreferent] class.

Second, the coreference relation is not homogeneous: it is affected by a variety of linguistic parameters and different pairs may require very specific resolution strategies (for example, two mentions of "Washington" are likely to be coreferent, whereas two mentions of "it" are not).

Third, we need a manually annotated training corpus to induce a prediction function for a learning-based coreference resolution engine[2]. Inconsistencies of the training material directly affect the classifier, decreasing its performance. This makes a learning-based approach sensible to the data: on the one hand, it can better capture data-specific patterns (and is therefore more robust, see above), but, on the other hand, it is affected by the annotation quality. The moderate quality of the manually annotated material can be explained not only by inaccuracies of the annotators (typos, etc), but also by the intrinsical difficulty of the task. The inter-annotator agreement for the MUC-7 data, used in our study, lies in low eighties (Hirschman et al., 1997).

[2]We do not discuss unsupervised or semi-supervised learning algorithms in this thesis.

Knowledge-poor vs. Knowledge-rich algorithms. The first coreference resolution systems relied extensively on hand-coded domain-specific knowledge. For example, Wilks (1973) suggests using a precompiled set of semantic constraints for pronoun resolution:

(7) $[\text{John}]_1$ took $[\text{the cake}]_2$ from $[\text{the table}]_3$ and ate $[\text{it}]_{4,ante=2}$.

The pronoun "it" has three possible antecedents: "John", "the cake" and "the table". The first candidate, "John" can be filtered out by gender agreement constraints (a list of person names marked for gender is needed) or by contra-indexing syntactic patterns (a list of patterns and a parser are needed). The system of Wilks (1973) relies on a knowledge base to find out that *cake* is *eatable* whereas *table* is not and conclude that "it" is coreferential with "cake".

Knowledge-intensive approaches received a lot of criticism already in the late eighties (Carbonell and Brown, 1988) for their very low flexibility. Even if we collect a reasonable knowledge base for a limited domain (e.g., a list of *eatable* objects), we cannot re-use it for other corpora, and creating a large-scale general-purpose base of world knowledge seems unfeasible.

This criticism was reflected in the recent trend to rely on knowledge-poor algorithms for coreference resolution, especially for pronominal anaphora (Kennedy and Boguraev, 1996; Mitkov, 1998). Such approaches often do not even need a parser, operating on NP-chunks and their very shallow properties.

Knowledge-poor algorithms are very robust and easily portable across domains. They are not sensible to deficiencies of various manually created resources needed for knowledge-based coreference resolution. Such systems, however, can only handle the easiest anaphoric links – coreference is a complex phenomenon that cannot be fully accounted for by shallow techniques.

State-of-the-art systems show very similar performance figures: low sixties for NP-coreference and low eighties for pronominal anaphora (Cristea et al., 2002). We believe that knowledge-poor algorithms have already approached the upper bound for their performance: one cannot significantly improve such systems by changing the underlying resolution strategy or opting for a better machine learning module, without introducing more features.

In this thesis we investigate the usability of automatically acquired linguistic knowledge for coreference resolution. Unlike early knowledge-based studies, our system does not rely on hand-crafted domain-specific rule sets, but, on the contrary, employs state-of-the-art learning-based preprocessing modules (see Section 2.4) to acquire values for numerous linguistically motivated features. This allows us to build a flexible coreference resolution engine that can be quickly re-trained to cover other domains.

To summarize, we aim at creating a data-driven linguistically motivated coreference resolution algorithm. We opt for a learning-based approach to be able to build a highly scalable and robust system. State-of-the-art data-driven

algorithms, however, can only account for easy coreference links. One has to go for more sophisticated features to resolve difficult anaphors. This thesis offers a systematic investigation of such features.

2.2 Data

Any learning-based algorithm relies on a *training* corpus – a large and balanced collection of manually annotated examples. An additional *test* corpus should be reserved for evaluation purposes. The two datasets should represent the same domain(s) and genre(s) and follow the same annotation guidelines.

We rely on the MUC-7 corpus in our study – a collection New York Times articles (30 "dry-run" and 20 "formal" documents) annotated with coreference chains. The MUC corpus also contains a "training" part – a collection of raw texts (30 documents) and three annotated articles. "Dry-run" documents are mosty devoted to a single narrow topic, air crashes. "Formal" and "training" documents represent a variety of broader topics: politics, economy, science, and entertainment. Such diversity between different corpus sub-parts make the MUC-7 data very difficult for learning approaches.

We do not investigate unsupervised or semi-supervised learning techniques in this thesis and therefore we cannot extract any information from the unannotated "training" data. Throughout this study we will train our classifier on the 30 MUC-7 "dry-run" documents and refer to these data as our "training set". This may sound slightly confusing, but it is a common practice for the coreference resolution systems evaluated on the MUC-7 corpus and it allows us to directly compare our results with the state-of-the-art.

We use the three annotated "training" documents for our preliminary experiments and to assess the extraction quality for some features (see Experiments 2–10, and also Sections 4.8, 5.2). The articles are much longer than any of the 50 "dry-run" or "formal" documents and have more complex structure (for example, addressing multiple topics).

The MUC annotation schemes (two slightly different versions for the MUC-6 and MUC-7 datasets) have been adopted by many other annotation projects. MUC-style annotated coreference corpora are now available for different languages, for example, Dutch (Hoste and Daelemans, 2004), French (Azzam et al., 1998), German (Hartrumpf, 2001), or Romanian (Harabagiu and Maiorano, 2000). The MUC corpora have become a standard dataset for evaluating English coreference resolution systems, including the algorithms participated in the MUC competition directly (Baldwin et al., 1997) and many following approaches (Soon et al. 2001; Ng and Cardie 2002c; Yang et al. 2004, among others).

Despite such popularity, the MUC guidelines have received a lot of criticism for the underlying theoretical assumptions (van Deemter and Kibble, 2001)

and the scoring algorithm (Bagga and Baldwin, 1998a). A new dataset with a different annotation scheme, ACE, has been proposed recently (NIST, 2003). We briefly summarize and discuss the MUC-7 guidelines below.

Coreference links are marked by COREF tags within the text steam, augmenting the original New York Times SGML structure (DOCID, STORYID. . .):

(8) <DOCID> nyt960405.0312 </DOCID>
<STORYID cat=e pri=u> A9753 </STORYID>
<SLUG fv=tdt-z> BC-MUSIC-LOVE-BOS </SLUG>
<DATE> (</DATE>
<NWORDS> <COREF ID=62>04-05</COREF> </NWORDS>
<PREAMBLE>
BC-MUSIC-LOVE-BOS
<COREF ID=1>LOVE & ROCKETS</COREF> LAUNCH <COREF ID=22>ANGST</COREF> MINUS <COREF ID=26>MELODIES </COREF> (For use by New York Times News Service clients)
By JIM SULLIVAN
c.1996 The Boston Globe
</PREAMBLE>
<TEXT>
<p>
Generally, modern rock fans have the attention span of cats, but not this time. Not when it comes to <COREF ID=0 TYPE=IDENT REF=1> Love & Rockets, <COREF ID=2 TYPE=IDENT REF=0 MIN="trio"> the British trio that was <COREF ID=3 TYPE=IDENT REF=2 MIN= "offshoot">an offshoot of the early '80s goth band Bauhaus</COREF> and was last glimpsed on the charts in 1989</COREF></COREF>. OK, <COREF ID=4 TYPE=IDENT REF=3>they</COREF> don't reenter the pop world at the same level. <COREF ID=5 TYPE=IDENT REF=4>They</COREF> went on hiatus following <COREF ID=7 MIN="hit">an improbable <COREF ID=12>US</COREF> hit, <COREF ID=6 TYPE=IDENT REF=7>"So Alive,"</COREF></COREF> and were playing the summer shed circuit.

The relation is further specified with extra attributes to the COREF tag:

ID is a unique identificator assigned to each "markable". Markables can be nouns, pronouns, or named entities.

TYPE is a type of the marked coreference link. The MUC-7 corpus contains only annotated links of the type "IDENT" (roughly speaking, pairs of markables referring to the same object, see below), but the guidelines claim that additional types (e.g., set/superset) can be included within the same scheme.

REF is a pointer indicating the antecedent's ID. The REF attribute is used to signal a coreference link between two markables. Thus, the annotation in our snippet (8) suggests, for example, a link between the 6th and the 7th markables, "an improbable US hit" and "So Alive".

MIN attribute designates the minimal part of the markable to be annotated (see below).

STATUS is a rarely occuring argument indicating that the annotated relation is optional (STATUS=OPT) and a system should not get penalized for failing to reproduce the link.

The MUC-7 guidelines specify how to define the units to be annotated (markables, MIN) and what kind of relations to encode (TYPE). These decisions are motivated by the following criteria, listed in order of their priority:

1. Support for the MUC information extraction task;
2. Ability to achieve good (ca 95%) interannotator agreement[3];
3. Ability to mark text up quickly (and therefore, cheaply);
4. Desire to create a corpus for research on coreference and discourse phenomena, independent of the MUC extraction task.

Markables. The units to be annotated, "markables", include nouns, noun phrases, and pronouns. Named entites (as defined for the MUC NE-extraction task) are also considered noun phrases. WH-phrases ("[Which engine] would you like to use?") are not markables[4]. Bare nouns (prenominal modifers) are only marked if they form a coreference chain with at least one full NP. Gerunds ("[Slowing the economy] is supported by some Fed officials; it is repudiated by others") are not markables. Substring of NEs ("Equitable of [Iowa] Cos") are not markables either. Empty strings (for example, zero pronouns, as in "Bill called John and [∅] spoke with him for an hour") are not to be marked. Coordinate NPs and their individual conjuncts are markables (this makes the MUC-7 guidelines different from the MUC-6 scheme, not treating coordinations as markables).

Not all the possibly identifiable markables appear in the data. The annotators are instructed not to mark trivial coreference relations (each noun

[3]Hirschman et al. (1997) note that the actual inter-annotator agreement was in low eighties.

[4]All the examples below are taken from the MUC-7 annotation guidelines. It can be argued that an alternative linguistic analysis is possible for some cases, especially for the following example of a zero pronoun.

phrase is coreferent to itself): a description is only a markable if it is coreferential with some other description. This consideration significantly reduces the set of markables but makes it impossible for coreference resoluton systems to automatically recreate MUC-7 markables from raw texts (see Chapter 7 for related experiments).

The MUC-7 scheme instructs the annotators how to encode selected markables. Various coreference resolution systems may employ different parsing or chunking algorithms and therefore the guidelines have been designed to support the maximal flexibility in bracketing NP-like units. A system is allowed to generate any chunk that is a substring of the annotated markable[5] and includes its "head". The MUC "head" is defined as "the main noun" for common noun phrases and as the entire name for named entities.

The head is designated by the MIN attribute. The MUC scoring program aligns manually annotated markables with the chunks provided by the system. For example, the MUC-markable "<COREF ID=2 TYPE=IDENT REF=0 MIN="trio">the British trio that was an offshoot of the early '80s goth band Bauhaus and was last glimpsed on the charts in 1989</COREF>" can be aligned with "trio", "British trio", "the British trio", "trio that", etc.

Such flexible annotation for markables should allow to use any syntactic representation of the raw text data provided by the MUC committee and therefore make it possible to concentrate the efforts on the coreference relation proper. The dry-run subcorpus contains 2569 markables, and the test subcorpus – 1728.

Relations. The annotation scheme covers only the nominal identity coreference, IDENT, to preserve high inter-annotator agreement (Criterion 2). It does not account for such phenomena as clause-level coreference ("Be careful not to [get the gel in your eyes]$_1$. If [this]$_{2,ante=1}$ happens, rinse your eyes with clean water and tell your doctor") or other relations (for example, bridging anaphora, as in "I got on [a bus]$_1$ yesterday and [the driver]$_{2,ante=1}$ was drunk.").

There is no explicit definition of the IDENT relation in the MUC-7 guidelines. They only suggest that "the basic criterion for linking two markables is whether they are coreferential: whether they refer to the same object, set, activity, etc. It is not a requirement that one of the markables is "semantically dependent" on the other, or is an anaphoric phrase" (Hirschman and Chinchor, 1997).

The basic criterion is extended to cover difficult and sometimes controversial types of links. The annotators should, in most cases, mark as coreferent "bound anaphors" and the NPs that bind them (9), parts of appositions (10),

[5]The articles are ignored.

predicate nominals and their subjects (11), and functions and their (most recent) values (12):

(9) [Most computational linguists]$_1$ prefer [their]$_{2,ante=1}$ own papers.

(10) [Julius Caesar, [the/a well known Emperor,]$_{2,ante=1}$]$_1$..

(11) [Mediation]$_1$ is [a viable alternative to bankruptcy]$_{2,ante=1}$.

(12) [The stock value]$_1$ rose from \$8.05 to [\$9.15]$_{2,ante=1}$.

It can be argued (see Criticism below) that some of these relations are not instances of coreference proper.The MUC guidelines list the most problematic cases (for example, negated appositions, as in "Ms. Ima Head, never a great MUC fan") as exceptions that should not be marked. We refer the reader to Hirschman and Chinchor (1997) for details.

The IDENT relation induces a set of equivalence classes ("chains") among the markables: it is a symmetrical, transitive, and reflexive function. The annotators are free to choose *any* element in the correct chain to encode as an antecedent for a given markable. For example, the 4th and the 5th markables in (8), pronouns "they", are linked to rather unintuitive antecedents, "an offshoot" and "they". This makes the MUC data very different from corpora annotated for *anaphora* (for example, pronominal anaphora datasets), where the "most suitable" antecedent is to be picked.

The IDENT relation is not directional, although the attributes induce some ordering (from ID to its REF) to facilitate the annotation process and improve the readability of the data.

The dry-run subcorpus contains 1905 links, and the test subcorpus – 1311 (including 37 links marked as "optional").

Scoring. The MUC-7 dry-run corpus can be used as a training set for a learning-based approach or as a collection of relevant examples for a rule-based system. The 20 "formal" documents are reserved for evaluation. The performance of a coreference resolution engine is estimated by comparing its annotation of the test data ("response") with the manual annotation provided by the MUC-7 team ("key"). The comparison function (Vilain et al., 1995) is implemented in a scoring program included in the MUC-7 distribution package.

The definition and properties of the scoring algorithm are very important – the MUC score helps us distinguish between better and worse coreference resolution systems and indicates directions for future work. It is therefore essential to design as intuitive an evaluation function as possible.

The MUC F-measure is a harmonic mean of the precision and recall values assigned by a model-theoretic scoring algorithm (Vilain et al., 1995). The

scorer computes the recall value by comparing the output of a system to be evaluated ("response") with the manual annotation ("key"). It determines the minimal number of links that have to be added to the response to put all the markables, coreferent according to the key, into the same chain. The precision value is computed by switching the roles of the key and the response. We describe the scoring procedure in details below.

The scorer first combines the markables shown in the key with those suggested in the response. Specific techniques have been developed to align syntactically different versions of the same NP (see above). The (combined) set of markables is then partitioned into equivalence classes with respect to the key and to the response annotation. Each coreference class is a transitive closure of some coreference chain (recall that coreference, as defined in the MUC guidelines, is a transitive, symmetrical and reflexive relation). Note that some coreference classes contain just one element: for example, if a markable M is only mentioned in the key, it will constitute a singleton class in the partition generated by the response.

Let M_{SR} be the (combined) set of markables, $S = \{S_1, S_2, .., S_n\}$ – the partition generated by the key, and $R = \{R_1, R_2, .., R_m\}$ – the partition generated by the response:

$$S_i = \{m_{i1}..m_{ik} : \quad \forall\ t, q \quad m_{it}, m_{iq} \text{ are coreferent in the key}\}$$

$$R_j = \{m_{j1}..m_{jl} : \quad \forall\ t, q \quad m_{jt}, m_{jq} \text{ are coreferent in the response}\}$$

$$\cup S_i = M_{SR} = \cup R_j$$

We can further split elements of S by intersecting them with members of R:

$$S_{ij} = S_i \cap R_j$$

Let $p(S_i)$ be a set of non-empty classes among $S_{ij},\ j = 1..m$. If we want to generate the class S_i from scratch (i.e. an empty response with no links suggested), we need $\mid S_i \mid -1$ links, for example, "resolving" each markable $m_{it} \in S_i$ to its predecessor $m_{i(t-1)},\ t = 2..\mid S_i \mid$. For a given response R, we have to add $\mid p(S_i) \mid -1$ links to fully reunite all the non-empty components S_{ij} of the chain S_i.

The recall error for S_i is therefore the number of missing links ($\mid p(S_i) \mid -1$) divided by the number of links in the key ($\mid S_i \mid -1$). The recall value for S_i is computed as

$$R_{S_i} = 1 - \frac{\mid p(S_i) \mid -1}{\mid S_i \mid -1} = \frac{\mid S_i \mid - \mid p(S_i) \mid}{\mid S_i \mid -1}.$$

The recall error for the whole partition S is computed by dividing the cumulative number of missing links for all the classes S_i by the total number

of the links in the key. The recall value for S is therefore

$$R_S = \frac{\sum_i(\mid S_i \mid - \mid p(S_i) \mid)}{\sum_i(\mid S_i \mid -1)}.$$

Consider the response, suggested by our system for the snippet (8):

(13) nyt960405.0312
A9753
BC-[MUSIC]$_a$-[LOVE]$_b$-BOS
(
[04-05]$_c$
BC-[MUSIC]$_{d,ante=a}$-[LOVE]$_{e,ante=b}$-BOS
[LOVE &]$_{f,ante=e}$ ROCKETS LAUNCH ANGST MINUS [MELODIES]$_g$
(For use by New York Times News Service clients)
By JIM SULLIVAN
[c.1996 The Boston Globe]$_h$
Generally, [modern rock fans]$_i$ have the attention span of cats, but not this time. Not when [it]$_{j,ante=h}$ comes to [Love & Rockets]$_{k,ante=e}$, [the British trio]$_l$ that was [an offshoot]$_{m,ante=l}$ of the early '80s goth band Bauhaus and was last glimpsed on the charts in 1989. OK, [they]$_{n,ante=i}$ don't reenter the pop world at the same level. [They]$_{o,ante=n}$ went on hiatus following [an improbable [US]$_q$ hit]$_p$, "So Alive," and were playing the summer shed circuit.

Figure 2.1 shows the combined set of markables M_{SR} with the key classes (solid lines) and the response classes (dashed lines)[6]. Note that the snippet is taken out of its context and therefore both the key and the response contain several singleton-looking chains – annotated markables, re-mentioned somewhere further in the document, for example, "04-05" ($M_{62=c}$).

The recall and precision values for our response are:

$$R = \frac{(6-4)+(2-2)}{5+1} \approx 0.3$$

$$P = \frac{(2-1)+(3-2)+(2-2)+(2-2)+(3-3)}{1+2+1+1+2} \approx 0.3$$

[6]Strictly speaking, we cannot align the key markable M_0 with the response markables M_k, bacause of the missing MIN argument in the manual annotation (cf. Example (8)). This is, however, an obvious typo and therefore we consider M_0 and M_k to be the same markable in our examples in this section.

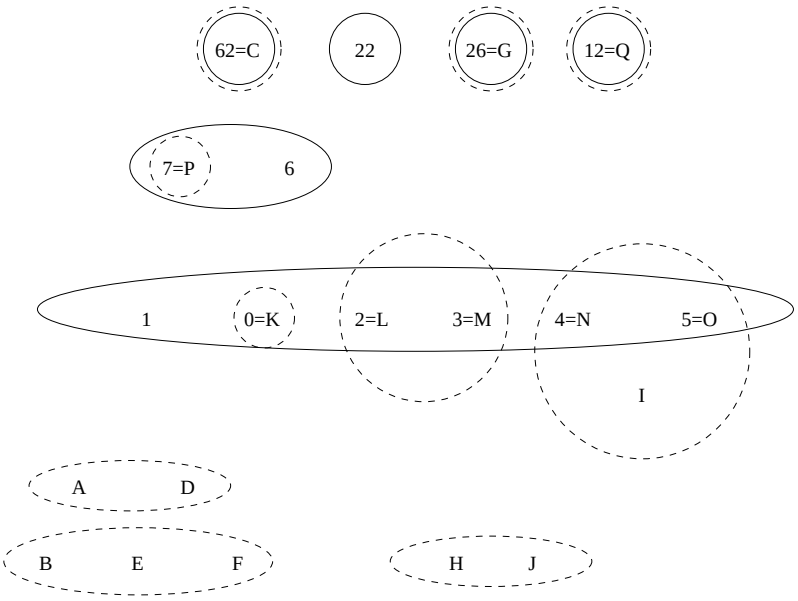


Figure 2.1: Comparing the MUC annotation (solid line) for Example (8) with the response produced by our system (dashed line).

Discussion. The MUC guidelines and data have been widely used by the computational linguistic community to create other corpora and evaluate coreference resoluton systems. Although they have established a standard for annotating coreference, a number of problems with the MUC approach have been discussed in the literature.

Van Deemter and Kibble (2001) have criticized theoretical assumptions of the MUC scheme, pointing out that the IDENT relation, encoded in the MUC corpora, goes far beyond coreference proper. They argue that it is important to distinguish between *coreference* and *anaphora*. Coreference is an equivalence relation – two descriptions are coreferent if and only if they refer to the same entity. Anaphora is an irreflexive, nonsymmetrical and non-transitive relation – an NP_1 is said to take NP_2 as its anaphoric antecedent if and only if NP_1 depends on NP_2 for its interpretation (Kamp and Reyle, 1993). The MUC guidelenes first claim that the IDENT relation should correspond to coreference, but then explicitly instruct the annotators to encode some cases of (non-coreference) anaphora as well. For example, bound anaphora should be annotated, according to the MUC scheme:

(14) [Every TV network]$_1$ reported [its]$_{2,ante=1}$ profits.

Van Deemter and Kibble (2001) argue that it is a very confusing decision: if "Every TV network" refers at all, then, presumably, it refers to the set of all TV

networks. Such annotations may lead to inconsistencies: if, for examlpe, (14) was followed by "[They]$_{3,ante=?}$ are now required to do this", the annotators would have to put "its" and "they"into the same chain.

Intensionality and predication are also problematic for the MUC scheme. The guidelines say at one point that "two markables should be recorded as coreferential if the text asserts them to be coreferential at ANY TIME":

(15) [Henry Higgins]$_1$, who was formerly [sales director of Sudsy Soaps]$_{2,ante=1}$, became [president of Dreamy Detergents]$_{3,ante=1}$.

The same guidelines instruct the annotators to pick "the most recent value" for relations involving "change over time":

(16) [The stock price]$_1$ fell from \$4.02 to [\$3.85]$_{2,ante=1}$.

Intensional and predicative descriptions are truly problematic and require specific annotation strategies. Van Deemter and Kibble (2001) suggest three possible remedies: annotating the present (instead of the most recent) value; introducing more complex referents[7]; and considering such expressions non-referring, leaving their analysis to the other tasks.

Separate processing of intensional and predicative descriptions would have both advantages and disadvantages for coreference resolution systems. On the one hand, intensionality and predication is de facto treated separately (mainly ignored) from coreference by most state-of-the-art algorithms. To our knowledge, most existing systems only rely on very simplistic patterns for appositions and copulas and do not attempt any analysis for such anaphors as "\$3.85" in (16) or even "president of Dreamy Detergents" in (15). Most coreference resolution systems would therefore benefit from a dataset, annotated for coreference proper, without noisy links involving intensional and predicative nominals.

On the other hand, predicate nominals often contain valuable knowledge that can help resolve other entities. Suppose that (15) is followed by a snippet mentioning "the president of Dreamy Detergents" once again. If we process predicate nominals completely separately, we have no information that help us link the second mention of "the president of Dreamy Detergents" to "Henry Higgins". Even if we integrate our module for predicative nominal into the main coreference resolution engine, we will still need much more sophisticated inference scheme here[8].

The MUC scoring algorithm (Vilain et al., 1995) has received criticism as well. Bagga and Baldwin (1998a) have outlined two shortcomings of the

[7]For example, "the stock price" in (16) can be analyzed as an *individual concept*, i.e. a function from "objects" to "prices", cf. (Dowty et al., 1981).

[8]See Section 4.7 for a discussion.

MUC scoring scheme: the algorithm does not give any credit for separating out singletons and considers all errors to be equal.

Singletons (referring markables, that do not participate in any coreference chain) are not annotated, according to the MUC guidelines, and therefore do not contribute to the score in any positive way. An incorrectly resolved singleton NP decreases the precision value, but a correctly identified and left unresolved singleton NP does not affect the system's precision or recall. Evaluated engines do not receive any direct benefit for correct extraction of markables. Most approaches therefore pay very few attention to their markables. We will see in Section 8.2 that deficiencies of the module for markable extraction account for 35% of our recall errors. A modified scoring scheme (and adjusted annotation guidelines), covering singleton NPs, could help create better pools of markables and thus improve the resolution quality.

Different kinds of spurious links receive the same penalty from the MUC scorer. Bagga and Baldwin (1998a) argue that some errors do more damage than others, at least for certain major tasks (for example, Information Extraction). Thus, a spurious link, merging two large chains into one should receive a bigger penalty than a link merging two small chains: the former makes more descriptions coreferent that should not be.

The MUC scoring algorithm is biased towards recall, although the definition ("precision and recall are computed by switching the roles of the key and the response") sounds symmetrical. Suppose we have two automatically constructed coreference chains, $R_a = \{a_1, a_2, a_3, \dots\}$ and $R_b = \{b_1, b_2, b_3, \dots\}$, and we know that a link between a_1 and b_1 is highly improbable. We can still obtain a better score by including the link $\{a_1, b_1\}$ into the response and thus merging two chains into one $R = R_a \cup R_b$. Consider the solution[9] proposed by our system for the TEXT part of (8):

(17) Generally, [modern rock fans]$_i$ have the attention span of cats, but not this time. Not when it comes to [Love&Rockets]$_0$, [the British trio]$_2$ that was [an offshoot]$_{3,ante=2}$ of the early '80s goth band Bauhaus and was last glimpsed on the charts in 1989. OK, [they]$_{4,ante=i}$ don't reenter the pop world at the same level. [They]$_{5,ante=4}$ went on hiatus following [an improbable US hit]$_7$, ["So Alive,"]$_6$ and were playing the summer shed circuit.

The annotators have suggested two chains for this snippet: $\{M_0, M_2, M_3, M_4, M_5\}$ and $\{M_7, M_6\}$. The response contains two different chains, $\{M_i, M_4, M_5\}$ and $\{M_2, M_3\}$. The recall value for this partition is $R_{old} = \frac{(4-2)+(1-1)}{4+1} = 0.4$.

[9] We retain the indices prosposed by the MUC annotators, adding singleton NPs to the response when necessary (M_0, M_6, M_7). The NP "modern rock fans" (M_i) was not shown in the key. See Example 13 and Figure 2.1 for the discussion of markable alignment in this snippet.

The precision value is $P_{old} = \frac{(1-0)+(2-1)}{1+2} = 0.66$. If we now deliberately spoil the response, by adding two unlikely (and indeed spurious) links $\{M_i, M_0\}$ and $\{M_i, M_2\}$, we can reunite the components of the initial chain and therefore obtain better scores: $R_{new} = \frac{(4-0)+(1-1)}{4+1} = 0.8$, $P_{new} = \frac{(5-1)}{5} = 0.8$. This observation suggests a trivial solution for the coreference resolution task as defined by the MUC committee – "always (or at least when in doubt) link everything". Although such an algorithm yields a good MUC score of 48% (two systems in the MUC-7 competition show worse results), it is obviously not informative and cannot be further used for Information Extraction or other tasks relying on coreference.

We will see in Chapter 7 that even a substancial improvement of the system's accuracy (by discarding automatically detected discourse new elements) does not necessary lead to a better MUC F-score.

We have mentioned the four crtitera behind the MUC guidelines. Two of them, "Support for the MUC Information Extraction Task" and "desire to create a corpus for research on coreference" suggest that the data should be suitable for applied studies, especially for training and evaluating coreference resolution engines. Some properties of the MUC corpus make it, however, difficult to use.

The data are very heterogeneous. First, the corpus is not balanced: it contains documents from various domains, different for the training and the testing parts. It is essential for a learning-based approach to select training and test instances from the same sample.

Second, semi-structured auxilliary parts (SLUG, DATE...) of New York Times articles are annotated. They have very specific structure and require different processing strategies than ordinary text. Note that our example response (13) contains much more mistakes in such auxilliary parts of the document, than in the main TEXT body.

Third, as argued by van Deemter and Kibble (2001), the IDENT relation goes far beyond coreference proper, incorporating related phenomena.

Such diversity in a relatively small dataset makes it problematic for machine learning. Rule-based systems are even stronger affected by this problem: too many hand-crafted heuristics are required to cover all the different phenomena addressed in the MUC coreference task.

The guidelines contain no formal definition of the most important MUC concept, IDENT. They only state that two markables are coreferent if and only if they refer to the same discourse object. There are virtually no instructions that can help annotators determine whether two markables are referential at all, and, if so, whether they indeed refer to the same object. Many decisions look therefore unintuitive or at least questionable:

(18) Search and rescue efforts were hampered Friday by [rain and heavy seas]$_1$...

> "[The weather]$_{2,ante=1}$ is not playing in our favor," said Blair Thomas, public affairs officer for Coast Guard.

It is intuitively unclear, whether the descriptions in (18) refer to some specific weather conditions in a particlular place. Provided the markables describe specific weather conditions, it is virtually impossible to determine whether they refer to the same object – we cannot determine when the second sentence was uttered and, even if we could, we have no formal algorithm for comparing the weather on, for example, Saturday and Friday. Even if it was raining on both Saturday and Friday, we can hardly consider $\{M_1, M_2\}$ an intuitive link.

Such examples inevitably decrease the MUC-7 inter-annotator agreement and make the data less suitable for machine learning.

Following van Deemter and Kibble (2001), we think that the MUC-7 definition of coreference should be revised and made more uniform and formal, if we want to create reliable systems.

To summarize, we rely on the MUC-7 data throughout this thesis. The MUC corpus has become a standard in the coreference resolution community. It is a small collection of New York Times articles annotated with coreferential nominal descriptions. Only the identity coreference is adressed. We have highlighted several problems with the theoretical assumptions of the MUC project and its scoring algorithm. We will see in Section 8.2 how these problems affect the performance level of our system.

2.3 Baseline

In this section we describe one of the first and the most successful learning-based approaches to the coreference resolution task – the algorithm proposed by Soon et al. (2001). This system has been used as a reference point by most following studies on coreference. The resolution strategy advocated by Soon et al. (2001) has been adopted by most state-of-the-art systems, although some improvements have been suggested (see, for example, (Ng and Cardie, 2002c) for a less local criterion for selecting antecedents).

The system of Soon et al. (2001) relies on a decision tree induced with the C5.0 learner. It achieves a performance level (60.5%) comparable to the best knowledge-based systems participated in the MUC-7 competition[10].

The system work as follows. It starts by extracting markables via a pipeline of language processing modules. Training instances are then created for appropriate pairs of markables (see below) and submitted to the C5.0 learner. The induced decision tree is used to generate the response annotation for the testing corpus: potential pairs of coreferring markables are submitted to the classifier, which decides whether the two markables actually corefer.

[10]No learning-based systems took part in the competition.

Features. The classifier relies on just 12 simple features, describing pairs (M_i, M_j), where M_i is a (candidate) anaphor and M_j is some preceeding markable:

1. `DIST` (0,1,2,...) encodes the distances (in sentences) between M_i and M_j.
2. `I_PRONOUN` returns true if and only if M_i is a pronoun.
3. `J_PRONOUN` returns true if and only if M_j is a pronoun.
4. `STR_MATCH` returns true if and only if M_i and M_j have the same surface form after stripping off the determiners.
5. `DEF_NP` returns true if and only if M_j is a definite NP (starts with *the*).
6. `DEM_NP` returns true if and only if M_j is a demonstrative NP (starts with *this*, *these*, *that*, or *those*).
7. `NUMBER` returns true if and only if M_i and M_j agree in number.
8. `SEMCLASS` returns true if M_i and M_j are semantically compatible (according to a simple IS-A hierarchy of semantic classes, cf. Section 5.2), false if they are incompatible, and unknown if the system fails to determine semantic classes for the markables.
9. `GENDER` returns true if M_i and M_j agree in gender, false if they disagree in gender, and unknown if the gender of either M_i or M_j cannot be determined.
10. `PROPER_NAME` returns true if and only if M_i and M_j are both proper names.
11. `ALIAS` returns true if and only if M_i and M_j are proper names and M_i is an alias of M_j or vice versa. The alias module works differently depending on the named entity type.
12. `APPOSITIVE` returns true if and only if M_j is in apposition to M_i.

Most features of Soon et al. (2001) can be re-implemented straightforwardly and require minimal knowledge. For example, we can determine whether a markable is a pronoun (`I_PRONOUN`, `J_PRONOUN`) without even a full-scale tagger, by looking it up in a short precompiled list of pronouns. Some features, however, are more application-specififc: thus, Soon et al. (2001) consult the WordNet database to determine and compare semantic classes of markables (`SEMCLASS`). The implementation for the `ALIAS` feature relies on hand-coded heuristics.

Generating training data. The 12 features are used to induce a decision tree (training) and then apply it for resolution (testing). We need additional pre- and post-processing steps to convert the MUC data, annotated with *chains*, into the C5.0 format, *feature vectors*, and back.

Training instances are generated as follows. First, candidate anaphors are selected by intersecting the pool of anaphors suggested by the MUC annotators (that is, all the markables with a non-empty REF value) with the pool of automatically extracted markables. If a MUC anaphor has not been identified successfully by the pipeline for markable extraction, the system cannot determine values for some features (for example, `GENDER`) and therefore such markables do not contribute to the training data.

Consider again our Example (8). The snippet contains six manually annotated anaphors: "Love&Rockets, the British trio that was an offshoot of the early '80s goth band Bauhaus and was last glimpsed on the charts in 1989" (M_0), "the British trio that was an offshoot of the early '80s goth band Bauhaus and was last glimpsed on the charts in 1989" (M_2), "an offshoot of the early '80s goth band Bauhaus" (M_3), "they" (M_4), "They" (M_5), and "So Alive" (M_6). Our system[11] has not recognized "So Alive" as a markable and, therefore, M_6 does not contribute to the training pool.

Positive training instances are created by pairing each candidate anaphor with its closest antecedent. Coming back to our example, we cannot generate any positive instances for M_0, because the only preceeding markable in its manually annotated chain, M_1, has not been recognized by our preprocessing modules. The other markables contribute the following positive instances (the anaphor is always shown first): {M_2, M_0}, {M_3, M_2}, {M_4, M_3}, and {M_5, M_4}.

Negative training instances are created by pairing each candidate anaphors with all the (automatically extracted) markables between the anaphor and its closest antecedent. This results in the following negative instances for our example:

{M_4, "1989"}
{M_4, "the charts"}
{M_4, "the early '80s goth band Bauhaus"}
{M_5, "the same level"}
{M_5, "the pop world"}

Note that we use automatically extracted units, and not only the MUC-7 markables as antecedents to generate negative instances.

Generating response annotation. Every automatically extracted markable is a potential anaphor at the testing step. It is paired with all the pre-

[11] We use our preprocessing modules for all the examples in this section, and, therefore, the selected instances may slightly differ from those generated by the system of Soon et al. (2001).

ceeding markables to form testing instances and generate feature vectors. The vectors are submitted to the classifier one-by-one, starting from the closest (rightmost) candidate and proceeding backwards. Once the classifier finds a positive instance, the corresponding markable is annotated as the antecedent and the system proceeds to the next candidate anaphor. If no antecedent is found, the candidate anaphor is left unresolved (and not annotated).

Consider the 5th markable, "they" from our example. The system generates one-by-one the following two instances and submit them to the classifier:
{M_5, "the same level"}
{M_5, "the pop world"}
Both candidates are rejected and the system proceeds with another pair, {M_5, M_4}. This instance is considered positive by the classifier, the corresponding link is added to the response annotation, and no more testing instances are generated for M_5. The system starts processing the next candidate anaphor, "hiatus".

To summarize, the system of Soon et al. (2001) is one of the first and the most successful learning-based approaches to the coreference resolution task. It re-casts the problem into two sub-tasks: making pairwise decision on possible coreference between markables and intellingenty merging these decision to create chains. This two-step view has been adopted and further elaborated by numerous studies. In this thesis we follow the same processing scheme and use the system of Soon et al. (2001) as a baseline.

2.4 External Modules

Any coreference resolution system relies on external linguistic modules. Even the most knowledge-poor algorithms need an NP-chunker to generate markables (and a sentence breaker to provide input for the chunker). Additional knowledge may be gained from a parser and an NE-tagger. Finally, learnin-based approaches rely on a machine learning module to induce a prediction function from the training data and apply it to the test data. In this section we briefly introduce the external linguistic resources used in our study.

We process each MUC-7 document with an SGML parser to extract its textual parts, PREAMBLE and TEXT. These parts are split into sentences as follows. Each line is a sentence in PREAMBLE. The TEXT body is submitted to a sentence breaker (Reynar and Ratnaparkhi, 1997). It is a publicly available maximum entropy-based algorithm for identifying sentence boundaries, trained on the Wall Street Journal corpus. The system operates on raw text and does not require any part-of-speech tags or domain-specific rules. Reynar and Ratnaparkhi (1997) report the accuracy level of 98.5-99% on the Wall Street Journal test data. We have observed several mistakes on the MUC data, but they have not affected the overall performance of our coreference resolution

system.

Extracted sentences (for PREAMBLE and TEXT) are then submitted to a parser (Charniak, 2000). This module assigns part-of-speech tags and determines the sentence structure, providing a parse tree. The parser's output is used for generating markables (Section 2.5) and for extracting values for our syntactic features (Chapter 4). The parser relies on a lexicalized Markov grammar approach, with a maximum entropy-based conditioning and smoothing model. Its publicly available version has been trained and tested on the Wall Street Journal data, achieving the performance level of 89.5-90.1% (average precision/recall for medium and short sentences correspondingly). We have observed a number of parsing errors on the MUC data, decreasing the performance of our main coreference resolution engine (see Sectuon 8.2).

The same sentences are simultaneously submitted to a named entity recognizer (Curran and Clark, 2003b). Its output is used for generating markables (Section 2.5) and for determining semantic classes for NEs (Section 5.2). The module is a maximum entropy-based tagger with an F-score of 85% reported for the CONLL data. It has been re-trained on the combined MUC-6 and MUC-7 NE corpus by James Curran and Stephen Clark to cover the MUC classification of named entities, but the performance figures for this experiment are not available. We discuss NE-tagging errors, affecting the main system's output, in Section 8.2.

We use the WordNet ontology (Miller, 1990) to obtain values for our semantic features. WordNet is a large publicly available lexical resource for English. The ontology is based on an IS-A forest of synsets. Each synset is an atomary word sense unit. For example, the noun *anaphora* is mapped to 2 synsets: "anaphora – using a pronoun or other pro-word instead of repeating a word" and "epanaphora, anaphora – repetition of a word or phrase as the beginning of successive clauses". Note that the second synset corresponds not only to the noun *anaphora*, but also to *epanaphora* – the mapping between synsets and words is a many-to-many correspondence. The current release of WordNet (version 2.1) contains 81426 synsets for 117097 nouns. We do not attempt any intelligent word sense disambiguation and always chose the first synset for each noun. This allows us to organize nouns in an IS-A hierarchy and thus compute hypernyms (for example, *repetition* is a hypernym of *anaphora*) and superconcepts, or semantic classes (*anaphora* is ABSTRACTION) for the head nouns of our markables. The WordNet ontology also shows additional relations between synsets (for example, *syllable* is a meronym of *word*). This information, however, is not used in this thesis: it would require functions for extensive sub-graph search and significantly slow down the system.

We need a machine learning module to induce a prediction function from pairs of [±coreferent] markables generated from the training data. The prediction function is a core component of any coreference resolution system and it is therefore essential to choose an appropriate machine learning module.

We have mainly used the SVM^{light} learner throughout this thesis. It is an implementation of the Support Vector Machines (SVM) learning algorithm (Vapnik, 1995), that has shown very promising performance for a variety of natural language processing tasks.

Support Vector learning has recently become very popular. On the one hand, it enables rather complex models but, on the other hand, it is simple enough to be analyzed mathematically within the Statistical Learning Theory framework (Vapnik, 1995). We only use the most simple linear SVMs throughout this thesis[12]. The (very brief) description below therefore only addresses linear SV learning.

One of the main concepts behind the SV learning is the *capacity* of a classifier. Burges (1998) explains it as follows: "for a given learning task, with a given amount of training data, the best generalization performance will be achieved if the right balance is struck between the accuracy attained on that particular training set, and the "capacity" of the machine, that is, the ability of the machine to learn any training set without error. A machine with too much capacity is like a botanist with a photographic memory who, when presented a new tree, concludes that it is not a tree because it has a different number of leaves from anything she has seen before; a machine with too little capacity is like the botanist's lazy brother, who declares that if it's green, it's a tree".

The Statistical Learning theory introduces a formal measure of capacity, the Vapnik-Chervonenkis (VC) dimension, and establishes a relation between the expected risk (expectation of the test error) and the VC dimension.

A simple linear SVM model addresses a task of pattern recognition – mapping points x in $\mathbf{R}^n$ ("patterns") to $\{+1, -1\}$ – by separating them with a hyperplane ($a * x + b = 0 : \quad a \in \mathbf{R}^n,\ b \in \mathbf{R}$). It can be formally shown that the *maximal margin* hyperplane has the lowest expected risk and, therefore, yields the best generalization (among the hyperplanes). Consider a simple separable problem on Figure 2.2: to separate black points (b_i) form white points (w_j) in $\mathbf{R}^2$ we have to find a line l maximizing the margin $m = \min_i dist(b_i, l) + \min_j dist(w_j, l)$. The maximal margin hyperplane is a solution of a quadratic optimization problem that only depends on dot products between patterns.

Recall that we follow the approach of Soon et al. (2001) and re-cast the coreference resolution problem into a combination of pattern recognition and clustering (see Section 2.3). Appropriate pairs of markables are encoded as patterns (feature vectors) and mapped to $\{+1, -1\}$ ($\pm$ coreferent) and a machine learner is used to induce such mapping from the manually annotated training material. This task formulation allows us to plug the SVM^{light} module directly into our coreference resolution engine.

[12] We have tried cubic kernels in a pilot experiment, but have not observed any significant improvement.

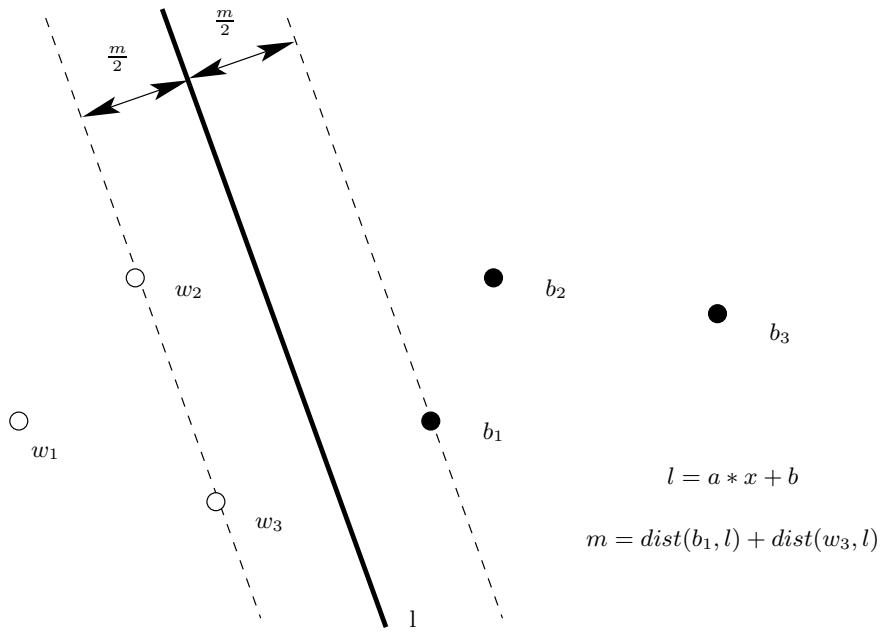


Figure 2.2: Hyperplane separation for a toy problem

We use additional publicly available machine learners in our final experiments: Ripper and Slipper (an information gain-based rule-induction system, cf. (Cohen, 1995)), C4.5 (a decision tree learner, cf. (Quinlan, 1993)), and MaxEnt (maximum entropy modeling with generative iterative scaling, cf. (Le, 2004)). We evaluate our system with these learners to confirm the findings obtained with the SVM^{light} module. The advantages and disadvantages of these learning modules are briefly discussed in Section 8.1.

To summarize, coreference is a complex phenomenon and every system needs a number of preprocessing modules to obtain linguistic knowledge relevant for the task. We rely on several maximum entropy-based systems (Reynar and Ratnaparkhi, 1997; Charniak, 2000; Curran and Clark, 2003b) and a manually created ontology (Miller, 1990) to generate markables and obtain values for our features. We use the SVM^{light} learner to induce a prediction function and classify pairs of markables as [±coreferent]. Additional learners have been evaluated in our final experiments.

2.5 Markables

The ultimate goal of any coreference resolution algorithm is to partition nominal descriptions (markables) in an arbitrary document into classes (chains), corresponding to discourse entities. It is therefore essential to have an extensive list of such descriptions to be processed.

Most studies on coreference do not pay attention to markable extraction. Some systems (Harabagiu and Maiorano, 2000) use exactly the units suggested by the annotators. This significantly simplifies the task (by eliminating spurious markables and reducing the search space) and thus allows to obtain better results. We will see below that deficiencies of the module for markables extraction account for 35% of our recall and 18% of our precision errors (cf. Section 8.2).

We use the following procedure to automatically generate markables from the raw MUC-7 data. We separate auxiliary semi-structured parts of the corpus from text parts by identifying New York Times SGML structure of the processed fragment: PREAMBLE and BODY are texts, whereas DOCID, STORYID, SLUG, NWORDS, DATE, and TRAILER are NYT-specific formatted strings.

Textual parts of the document are segmented into sentences (Reynar and Ratnaparkhi, 1997). We parse the obtained sentences (Charniak, 2000) and extract named entities (Curran and Clark, 2003b). We then merge the outputs of the parser and the NE-tagger to create a list of markables as follows:

1. Named entities are considered markables if and only if they correspond to sequences of parsing constituents. Any partial overlap between NEs and parsing units is prohibited: for example, the sentence "For use by New York Times News Service clients" in (8) has been misanalyzed by the parser as "[For use by New York Times] [News Service clients]" and the system has therefore discarded the NE candidate markable "New York Times News Service".

2. Possessive pronouns are markables if they are not parts of named entities.

3. Noun phrases (including non-possessive pronouns) are "candidate markables" if they are not parts of named entities. The set of candidate markables is filtered to eliminate pairs of NPs with the same head noun[13] – embedding NPs are discarded. The remaining NPs are added to the set of markables. For example, "the British trio that was an offshoot of the early '80s goth band Bauhaus and was last glimpsed on the charts in 1989" and "the British trio" are both represented by the embedded description "the British trio". The selected NP-markables are additionally aligned with the (already extracted) named entities if they share the same last word. For example, "guitarist Daniel Ash" and "Daniel Ash" get aligned and become one markable.

This procedure results in a pool of "basic" units: a markable can be embedded in another markable only if they have different head nouns, for example, in possessive constructions ("[[David Essex]'s granddad]") and coordinations

[13] cf. Section 4.4 for the algorithm for determining heads.

("[[the glam-rock suggestiveness], [the anguish] and [the angst]]"). It is essential for a coreference resolution system to operate on such basic descriptions, as they are less sensible to parsing errors than full NPs and can therefore be reliably extracted and quickly compared to each other. It is nevertheless important to pay attention to the context of a markable (including embedding noun phrases).

Auxiliary parts of a document (SLUG, DATE, NWORDS, and TRAILER) are split into words and then each word is considered a markable. Date-specific descriptions (DATE and TRAILER) are treated separately – each date-formatted substring is a markable. More elaborated processing of semi-structured data (see Example (8) for the typical values of these fields) lies out of the scope of this thesis.

We finally obtain a relatively large pool of descriptions. Our system has, for example, extracted 3251 markables for the test data, compared to 1728 markables suggested by the MUC annotators.

2.6 Summary

In this chapter we have introduced the methodology followed throughout the thesis. We advocate a learning-based algorithm, relying on a rich feature set. The motivation behind such an approach is discussed in Section 2.1.

It is essential for a data-driven approach to rely on a high-quality corpus. Our study is based on the MUC-7 dataset. This is a standard corpus for building and evaluating coreference resolution systems. It is briefly described in Section 2.2.

We use as a baseline the system of Soon et al. (2001) – one of the most successful learning-based approaches, evaluated on the MUC data. The algorithm is summarized in Section 2.3.

Every coreference resolution system needs a variety of preprocessing modules. We introduce the external resources, used in our study, in Section 2.4.

Finally, Section 2.5 shows our algorithm for extracting markables – basic units for any coreference resolution engine to operate on.

We process MUC documents with a number of linguistic modules (Section 2.4) to generate markables (Section 2.5) and select appropriate pairs of markables as training and testing instances (Section 2.3). In the following chapters we investigate linguistic properties interacting with coreference and encode the relevant information to create feature vectors for the selected instances. We start with the most shallow name-matching parameters (Chapter 3) and then proceed to syntactic (Chapter 4), semantic (Chapter 5), and discourse (Chapter 6) knowledge.

Chapter 3

Surface Similarity: Name Matching

In the present and the following chapters, we concentrate on the usability of different kinds of linguistic knowledge for statistical coreference resolution. We start with the most shallow information — surface similarity of nominal descriptions. In this chapter we review the existing matching strategies and propose a novel methodology. We then investigate, to what extent these shallow techniques may help a coreference resolution engine.

Co-referring descriptions often have a similar surface form. The same string can be simplified or repeated as it is. The task of name-matching consists in identifying variants of the same name. Consider again our Example (1) repeated below:

(19) One reason Lockheed Martin Corp. did not announce a full acquisition of Loral Corp. on Monday, according to [Bernard Schwartz]$_1$, [Loral's chairman]$_{2,ante=1}$, was that Lockheed could not meet the price [he]$_{3,ante=2}$ had placed on Loral's 31 percent ownership of [Globalstar Telecommunications Ltd]$_4$.
[Globalstar]$_{5,ante=4}$ plans to provide telephone service by bouncing signals off 48 low-orbiting satellites. But with no customers expected until 1998, the need for nearly $2 billion in investment and numerous competitors lurking in the shadows, [Globalstar]$_{6,ante=5}$'s prospects would not appear to be valuable to the average Lockheed shareholder. Still, [Schwartz]$_{7,ante=6}$ feels differently, and so now do many investors.

The same company, Globalstar, is mentioned three times within this small snippet: twice as "Globalstar" and once as "Globalstar Telecommunications Ltd". These mentions can clearly be analyzed as a simplified and a full version of the same description. In this chapter we propose shallow techniques, accounting for the most common variation patterns.

An advanced name-matching module may constitute a backbone for a full-scale coreference resolution. First, as we will see in Section 3.5, it can potentially resolve around half of the anaphors. Second, shallow surface similarity-based classifiers can be easily re-trained to cover other domains or, more important, other languages, where no reliable NLP tools are available. It should, however, be kept in mind that even the most accurate name-matching algorithm cannot account for all the types of anaphors. Thus, the same snippet (19) contains three anaphoric mentions of the same person, Bernard Schwartz, and only one of them, "Schwartz", can be successfully resolved with a name-matching algorithm.

Good matching techniques are especially important for proper-name coreference. There are several reasons why one should pay extra attention to this sub-task. First, simply identifying all anaphoric links between named entities correctly, one can achieve around 30% recall and 100% precision in MUC-style set-based evaluation. This means that proper name pairs represent a high proportion of anaphoric links and are therefore important for accurate coreference resolution.

Second, coreference resolution engines are usually not stand-alone tools — they are integrated into various NLP applications. For example, a good coreference resolution module in a question answering system would allow to keep track of entities, collecting and combining information from different sentences in a text, or even from different texts (in a cross-document coreference setting). In this context, coreference links between named entities become very important: they bring together various facts about real world objects that are likely to be mentioned in a user's questions.

Third, accurate name matching can be helpful for other tasks. Borgman and Siegfried (1992) provide an overview of name-matching algorithms for database management, in particular, identifying and removing duplicate entries. Branting (2002) points out that good name matching is even more crucial for Legal Case-Management Databases. These systems should identify conflicts of interests, checking, for example,whether an attorney has a personal interest in the outcome of a case. This task is typically solved by comparing a *conflict file* (list of all the affected parties) with the names of attorney and judge candidates. Wang et al. (2004) propose a name-matching algorithm for detecting fraud cases in databases entries, for example, forged identity cards used for criminal purposes.

In the next two sections we describe the relevant studies and highlight challenging name-matching problems. Section 3.3 introduces our methodology for assessing surface similarity. Section 3.4 describes our name-matching features. Finally, the approach is evaluated on named entities and full-scale coreference resolution in Section 3.5.

3.1 Related Work

The name-matching task has long been explored by different scientific communities, resulting in a number of approaches:

Record matching. This approach has been mostly advocated by database studies. It relies on external evidence for name-matching, comparing not only the names themselves, but the whole corresponding database entries. This involves, for example, searching for an entry in the U.S. census data by matching not only a person's name, but also the corresponding birth date or the social security number (Winkler, 1999). A similar approach can be used for free texts as well. Bagga and Baldwin (1998b), Fleischman and Hovy (2004), and Mann and Yarowsky (2003) describe different systems for cross-document person name resolution based on external clues: first, they encode names' contexts in each document (as bag-of-words or N-grams) and then use probabilistic models to cluster the names into different "persons".

Distance-based matching. Various researchers (Bilenko and Mooney 2002; Cohen et al. 2003, among others) have tried to design edit distance-based metrics for name matching. These methods are very important for databases with poor internal structure. Virtually none of the previous approaches to the coreference resolution task include any specific string matching metric. An exception is a system described in (Strube et al., 2002b), where two approximate matching strategies have been tested.

Language-specific matching. Patman and Thompson (2003) propose a multi-cultural name-matching approach: they first guess the origin of the name and then apply culture-specific rules. For example, their system prohibits matching of "Khalid bin Jamal" to "Khalid abu Jamal": it classifies these names as Arabic, parses them accordingly and applies the rule saying that "A bin B" ("A, son of B") and "A abu B" ("A, father of B") are incompatible. This approach, however, implies a lot of handcoding. There have been several database name-matching approaches of this kind designed predominantly for English names, for example, the systems described in (Borgman and Siegfried, 1992). Some coreference resolution engines rely on (very simple) language-specific matching techniques. Soon et al. (2001) use the *weak string identity* measure, comparing two strings after stripping off determiners. Bontcheva et al. (2002) have several heuristics for identifying identical names in English, for example, matching phrases with and without prepositions: "University of Sheffield" and "Sheffield University".

3.2 Challenges for Name-Matching Algorithms

Name matching can be defined as determining whether two strings are variants of the same (proper) name:

(20) "Satellites give us an opportunity to increase the number of customers we are able to satisfy with the McDonald's brand," said McDonald's Chief Financial Officer, *Jack Greenberg*. [...] *McDonald's Greenberg* is quick to point out that the company opened about 1,800 free-standing restaurants last year, 50 percent more than in 1994, partly the result of a cost-cutting program. [...] "A few years ago we were only opening a couple of hundred," *Greenberg* said. "With the low-cost approach, we are able to open more."

For this simple example, an accurate name-matcher should suggest that "Jack Greenberg", "McDonald's Greenberg", and "Greenberg" form one coreference chain.

The study of Nenkova and McKeown (2003) suggests several regular patterns for the first and subsequent mentions of PERSON names in newswire texts, describing possible changes in name realization (full, first, last, or nickname) and modification (title, premodifiers, postmodifiers). Unfortunately, these patterns can be used only for English names (`First Name + (Middle Names) + Last Name`). Names of non-English origin and Named Entities other than PERSON are both problematic for an approach of this kind.

The main difficulty with non-English names comes from their structure: it might be very difficult to distinguish between the given and the family name. For example, Chinese or Korean names may be written in more or less any order: the traditional format requires the reversed order (the family name comes first), but many people prefer the Western style (the given name comes first), especially when uttering their name in a European language. Thus, Tony Leung, a Chinese actor, is mentioned in the IMDB database[1] as "Tony Leung Ka Fai", "Kar Fai Leung", "Tony Ka Fai Leung", and "Tony Leung"

In some countries people have just one name followed or preceded by a title: the Indonesian linguist Agus Salim has only one name, "Salim", and the title "Agus".

Patman and Thompson (2003) show even more culture-dependent examples of different name styles (see there for details). This makes the rewriting approach suggested by Nenkova and McKeown (2003) hardly feasible, once the data contain foreign names.

Other types of Named Entities pose additional difficulties, Although the general simplification tendency noted by Nenkova and McKeown (2003) holds

[1] `http://www.imdb.com`

for some cases of NEs, for others it is not predictive. For example, "Panasonic Toyota Racing" is the same as "Toyota", but not "Panasonic", whereas "Minardi Cosworth F1 Team" is the same as "Minardi", but not "Cosworth". Finally, some proper names can be simplified through abbreviation or compression: both "JFK" and "Kennedy" stand for "John F. Kennedy", but only "FAA", and not "Administration" stands for "The Federal Aviation Administration".

Another problem is data inconsistency: the same name may be written differently. This is a very important issue mainly for cross-document coreference, but even within one document we may find various irregularities. They can be divided into the following groups:

Different modifiers. The first mention of a name is very likely to be modified (0.76%, (Nenkova and McKeown, 2003)). At each next mention the name can still be modified (44%) or not (56%), until the simplest (non-modified) version is chosen. However, a name may have different modifiers throughout the document. For example, one might start with "Dr. Jones" and then continue to "Mrs. Jones". It is hardly possible to compile a list of compatible and incompatible modifiers: "Dr. Jones" and "Mr. Jones" can be instances of the same name, whereas "Mr. Jones" and "Mrs. Jones", as well as "his Toyota" and "my Toyota" cannot. Finally, "Jones Co" and "Jones Inc" still can be instances of the same name, one of them being simply a mistake.

Different auxiliary characters. Named entities of some NE-classes, in particular, PRODUCTs, often contain auxiliary characters. These symbols are usually not important and could be removed: "Boeing-747" is the same name as "Boeing 747". Unfortunately, this rule is not universal — "C", "C++", and "C#" are names of different objects.

Spelling variations. Even for languages with alphabetic scripts (such as Arabic or Cyrillic), there exist various national transliteration standards. For example, the last name of Vera Dushevina, a Russian tennis player, can be spelled as "Duševina" (GOST transliteration), "Douchevina" (French-style transliteration, old standard), "Dushevina" (English-style transliteration, new standard). It can also be written as "DyweBuHa" (so-called Volapuk system) or "DooSHEHveenuh" (American-style pronunciation-based system). For languages with non-alphabetic scripts, transliteration may result in more variants. These variants are not spelling mistakes but different correct transliterations.

If we take typos in account, both English and foreign names can potentially be spelled in much more ways. Table 3.1 shows the 10 most popular spellings of "Quaddafi" and "Britney Spears" on the web.[2]

[2]These statistics are collected by inspecting the data provided in `http://`

Quaddafi		Britney Spears	
Spelling	# pages	Spelling	# pages
Gaddafi	120000	Britney Spears	7420000
Qaddafi	79000	Brittany Spears	166000
Gadhafi	76000	Brittney Spears	123000
Kadhafi	14300	Britany Spears	55900
Kadhafi	14300	Britny Spears	18000
Kaddafi	5640	Britteny Spears	7260
Gadafi	3960	Briney Spears	6040
Qadafi	3780	Brintey Spears	5740
Al-Gaddafi	973	Brittny Spears	4950

Table 3.1: 10 most popular spelling variants for "Qaddafi" and "Britney Spears" on the web, with the corresponding number of pages in English, as indexed by Google.

Antecedent	Anaphor
MCDONALD'S	McDonald's
The Federal Aviation Administration	FAA
F-14	F14
CHINA'S Foreign Trade Minister Wu Yi	Ms Wu
Dan Goldin	Golden

Table 3.2: Examples of problematic cases for coreference resolution of proper names.

Well-known names. Some instances of (mostly very famous) names, in particular, names of LOCATIONs, can undergo significant changes in different languages, resulting in such pairs as "Munich"-"München" or "Aquis Grana"-"Aachen". Most documents use English variants of these names, but a few, for example, World Atlases, opt for national variants. Unlike the cases listed above, variants of famous names are very irregular and therefore problematic for an automated approach, making a lexicon-lookup the most preferable resolution strategy.

These inconsistencies occur even in relatively clean data. Table 3.2 shows several examples of coreferring proper names, demonstrating problematic cases in the MUC corpus. They include spellings variations w.r.t. the register and hyphenation ("McDonald's", "F-14"), abbreviations ("FAA"), foreign names ("Ms Wu") and typos ("Golden").

`www.google.com/jobs/britney.html` and `http://www.ecom.arizona.edu/ISI/2003/resources/presentation/NSF\_NIJ\_PRES.PPT`.

We have developed a set of name-matching techniques adjustable, in contrast to rule-based approaches, to a specific corpus: for example, if the data contain a lot of inconsistently spelled foreign names, the classifier will mostly rely on approximate matching, whereas for English names it will rather use substring selection and exact matching.

3.3 Computing Surface Similarity

State-of-the-art coreference resolution systems commonly employ a very simple matching algorithm, relying on exact string comparison. Several studies (Soon et al., 2001; Strube et al., 2002b; Bontcheva et al., 2002) show that it can be improved in many different aspects. We combine the ideas proposed in the literature with our own matching strategies. This results in a relatively large set of features and feature groups requiring comprehensive evaluation experiments to assess which techniques work better and why.

The most commonly used features for comparing the surface strings of two coreference candidates are the following:

- `same surface:` 1 if an anaphor and an antecedent have exactly the same surface form, 0 otherwise.

- `same head:` 1 if the head noun of an anaphor and an antecedent are exactly the same tokens, 0 otherwise. Head noun is defined as the last noun of a given NP (complex NPs are discarded).

- `contain:` 1 if an anaphor is a substring of an antecedent, 0 otherwise.

In our experiments we use two combinations of these features as baselines.

We next describe the different techniques and sub-algorithms we propose for the name-matching task. Combinations of the sub-algorithms are used in our system to compute values for numerous features. Several techniques are language specific, whereas others are relatively language independent (possibly applicable without modifications to any language with an alphabetic script). Some algorithms are very fast, some (parser-based) require more processing, and some (Internet-based) are time-consuming. We evaluate "relatively language independent" and "fast" settings in Section 3.5.

We decompose our problem into three novel sub-tasks:

1. normalization;

2. specific substring selection;

3. matching proper.

normalization function	normalized string
no normalization	that F-14
`no_case`	that f-14
`no_punctuation`	that F14
`no_determiner`	F-14
`no_determiner&no_case`	f-14
`no_punctuation&no_case`	that f14
`no_determiner&no_punctuation`	F14
`no_determiner&no_punctuation&no_case`	f14

Table 3.3: Different normalized forms for "that F-14".

So, one can, for example, compute minimum edit distance (*matching function*) between the down-cased (*normalization*) last nouns (*substring*) of an anaphor and an antecedent. The resulting value can be used as a similarity measure between the two.

Below we describe three classes of algorithms that we have implemented in order to tackle these three sub-tasks.

Normalization. The same name may be spelled differently throughout a text. For example, the headers of the MUC-7 documents are capitalized ("MCDONALD'S"), whereas the text bodies are not ("McDonald's"). The same noun phrase can sometimes be used with different determiners ("a/the restaurant"). We have investigated several normalization strategies to unify the surface form of such phrases.

We have tested three normalization functions: `no_case`, `no_punctuation`, and `no_determiner`. The first one transforms a string into lower-case format. The second one strips off all punctuation marks and other auxiliary characters (for example, "-" or "#"), and the last one strips off determiners. The first two normalization techniques are relatively language independent. The third one is obviously language dependent: we need an exhaustive list of determiners to perform this operation. However, we believe that such a list can be compiled very quickly for any particular language. One can combine these functions sequentially, producing complex normalizing algorithms, for example, `no_case&no_determiner`. Finally, one can use several normalizations for the same markable to compute values for different features. Table 3.3 shows different possible normalizations for the "that F-14" string.

Substring selection. One could argue that some words in a name are more informative than others. One could, therefore, instead of matching whole strings, compare only their most representative parts. For example, matching

"CHINA's Foreign Trade Minister Wu Yi" against "Ms Wu" becomes much easier once we know that "Wu" is the main part of both names. We have investigated several algorithms for automatically selecting the most informative words in a proper name:

head: this algorithm outputs the last noun of an NP string. It requires a parser (or at least a tagger) and is therefore highly language-dependent. We describe our algorithm for extracting the head of a markable in Section 4.4.

last: this is a simple modification of the `head` algorithm. It outputs the last word of an NP string. Although it is probably less accurate, it is faster and relatively language-independent. Its usability as a name-matching predictor may, of course, vary for different languages.

first: this is a counterpart of `last`. it outputs the first word of an NP string. Although it does not seem to be as useful as the previous techniques, we might need this algorithm to be able to resolve such coreference links as ("John Smith", "John") and ("Lockheed Martin Inc", "Lockheed"). A modified version, `firstnotitle` accounts for such coreference links as ("Mr Smith", "Smith") by stripping of title descriptors ("Mr(s), "Dr" etc).

rarest: this algorithm outputs the least frequent word of the NP string. It works as follows: each word is sent to the AltaVista search engine and then the one that gets the lowest count (number of pages worldwide written in English) is returned. Using this strategy we can account for tricky matches such as "Panasonic Toyota Racing" – "Toyota" or "Tony Leung Ka Fai" – "Mr. Leung". In particular, for personal names this can be seen as an automatic strategy for determining family names, avoiding expensive culture-specific handcoded rules. This algorithm is language independent, but very time-consuming. The speed depends on the selected search engine and the connection properties.

It does not make any sense to combine these algorithms sequentially, as they always output one word. So, for example, `rarest(first(NP))` is always identical to `first(NP)`. However, one can use several of them at the same time. For example, our `fast` configuration uses both the `first` and the `last` algorithms. One can also use no substring algorithms at all, comparing only the full strings of an antecedent and an anaphor. All the algorithms are illustrated in Table 3.4.

Matching. The above-mentioned algorithms help us create unified and simplified representations of two NP strings. We need a comparison function to assess the degree of similarity between these representations.

substring function	selected string
no selection	Lockheed Martin Corp.
`last_noun`	Corp.
`last`	Corp.
`head`	Martin
`first`	Lockheed
`rarest`	Lockheed

Table 3.4: Substring selection for "Lockheed Martin Corp."

A variety of string matching algorithms have been investigated for this purpose. Some of them do not impose any constraints on their input strings, whereas others make sense only when used together with particular substring selection algorithms.

exact_match is a boolean function on two strings. It outputs 1 if they are identical and 0 otherwise.

approximate_match algorithms are based on the minimum edit distance (MED) measure (Wagner and Fischer, 1974). Given two ordered sequences, the minimum edit distance between them is defined as a number of basic edit operations, needed to transform one sequence into the other. In this study we consider three standard edit operations: insertion, deletion, and substitution.[3] One can compute minimum edit distance between two strings either in symbols (`med_s`) or in words (`med_w`). For example, the distance between "New York" and "York" is 4 in symbols (*delete* "N", "e", "w", and " ") and 1 in words (*delete* "New"). When calculating distances in words we consider spaces to be word separators and do not take them into account. Punctuation symbols are, however, regarded as separate words (if `no_punctuation` normalization is not triggered).

The minimum edit distance measure has an obvious drawback: it does not reflect the length of a string — the distance between two similar, but very long NPs can be very high. So, we normalize our minimum edit distance (`MED`) values by the length of either an anaphor or an antecedent. The lengths are computed in symbols (`length_s`) or in words (`length_w`). Table 3.5 shows the formulae for our approximate matching functions and their values for the ("New York", "York") pair.

matched_part algorithms are generalizations of commonly used `contain` feature. They represent the size of the overlap between two NPs. The basic

[3]Branting (2002) takes into account an additional edit operation — reversal of pairs of adjacent letters.

matching function	formula	value for ("New York", "York")
`MED_s`	MED in symbols	4
`MED_s_anaph`	$\frac{MED\_s}{length\_s(anaphor)}$	1
`MED_s_ante`	$\frac{MED\_s}{length\_s(antecedent)}$	0.5
`MED_w`	MED in words	1
`MED_w_anaph`	$\frac{MED\_w}{length\_w(anaphor)}$	1
`MED_w_ante`	$\frac{MED\_w}{length\_w(antecedent)}$	0.5

Table 3.5: Approximate matching functions and their values for ("New York", "York").

(antecedent, anaphor) pair	abb1	abb2	abb3	abb4
(United States, U.S.)	1	1	1	1
(The Federal Bureau of Investigations, FBI)	0	1	1	0
(The Federal Bureau of Investigations, Bureau)	0	0	1	0
(Silicon, SILIC)	0	0	1	1

Table 3.6: Example values of the abbreviation functions.

`matched_part` algorithm computes the number of symbols/words two NPs share. The overlap values for ("New York", "York") are 4 and 1 correspondingly.

abbreviation algorithms compare an NP (full string) to the `head` of the other NP. The first algorithm (`abbrev1`) takes the initial letter of all the words in a string and produces a word out of them. This word is compared (by exact match) to the head of the second NP. The second algorithm (`abbrev2`) does the same, but ignores words, beginning with low-case letters: for example, it abbreviates "Federal Bureau of Investigations" to "FBI" and not "FBoI" as `abbrev1` would do. The algorithm `abbrev3` checks whether it is possible to split the head of the second NP into small units, so that each unit is a beginning (prefix) of a word in the first NP, and the prefixes come in the right order (the same as the order of the corresponding words). Finally, `abbrev4` proceeds in the same way, but does not allow empty prefixes. The first two algorithms represent the most commonly used abbreviations. The last two algorithms are more general, allowing for non-trivial ways of abbreviating. Table 3.6 shows some examples.

rarest(+contain) computes the `rarest` substring of an NP and then checks

lower_case	M contains lower-case letters (0,1)
cap_words	M is a sequence of words, starting with a capital letter (0,1)
upper_case	M contains upper-case letters (0,1)
digits	M contains digits (0,1)
alphas	M contains letters (0,1)
lower_case_h	M's head contains lower-case letters (0,1)
cap_words_h	M's head is a word, starting with a capital letter (0,1)
upper_case_h	M's head contains upper-case letters (0,1)
digits_h	M's head contains digits (0,1)
alphas_h	M's head contains letters (0,1)
rarest	the AltaVista count for the `rarest` word of M (1...n)
length_s	length of M in symbols (1...n)
length_w	length of M in words (1...n)

Table 3.7: Markable-level features

whether this particular word occurs in the other NP.

Again, one cannot combine these algorithms sequentially, but it is possible to use several of them at the same time, by computing values for different features.

Additional knowledge can be obtained by comparing syntactic properties of two named entities, for example, their number or NE-class (thus, not allowing matching PERSONs against LOCATIONs). In this section we are only interested in surface similarity, so we do not consider this information.

3.4 Features and Their Configurations

The algorithms introduced in the previous section help us build a set of name-matching features. We divide all these features into those that apply on the markable level (encoding properties of one NP) and those that apply on the coreference level (encoding properties of two NP, constituting a coreference candidate pair).

Markable-level features represent information about a single markable M. We compute their values for both an anaphor and an antecedent. Table 3.7 summarizes the features' definitions.

We do not use all these features in all our experiments. Thus, `lower_case_h`, `cap_words_h`, `upper_case_h`, `digits_h`, and `alphas_h` are activated only for experiments involving parsing. The `rarest_count` feature is used only in configurations when we need the `rarest` substring for matching.

Table 3.8: Name-matching features and their values for ("New York", "York"), `no_web_parser+no_normalization` configuration shown in italic.

Feature	Range	Example value
Anaphor's parameters		
lower_case(M_i)	0,1	1
cap_words(M_i)	0,1	1
upper_case(M_i)	0,1	1
digits(M_i)	0,1	0
alphas(M_i)	0,1	1
lower_case_h(M_i)	0,1	1
cap_words_h(M_i)	0,1	1
upper_case_h(M_i)	0,1	1
digits_h(M_i)	0,1	0
alphas_h(M_i)	0,1	1
rarest(M_i)	continuous	$816 * 10^6$
length_s(M_i)	continuous	8
length_w(M_i)	continuous	2
Antecedent's parameters		
lower_case(M_j)	0,1	1
cap_words(M_j)	0,1	1
upper_case(M_j)	0,1	1
digits(M_j)	0,1	0
alphas(M_j)	0,1	1
lower_case_h(M_j)	0,1	1
cap_words_h(M_j)	0,1	1
upper_case_h(M_j)	0,1	1
digits_h(M_j)	0,1	0
alphas_h(M_j)	0,1	1
rarest(M_j)	continuous	$816 * 10^6$
length_s(M_j)	continuous	4
length_w(M_j)	continuous	1

Table 3.8: Name-matching features and their values for ("New York", "York"), `no_web_parser+no_normalization` configuration shown in italic (continued).

Feature	Range	Example value
Pair's parameters		
exact_match(head(M_i,M_j)) [=same_head(M_i,M_j)]	0,1	1
exact_match(head(no_case(M_i,M_j)))	0,1	1
exact_match(M_i,M_j) [=same_surface(M_i,M_j)]	0,1	0
exact_match(first(M_i,M_j))	0,1	0
exact_match(first(no_det(M_i,M_j)))	0,1	0
exact_match(firstnotitle(no_det(M_i,M_j)))	0,1	0
exact_match(first(no_case(M_i,M_j))	0,1	0
exact_match(first(no_case(no_det(M_i,M_j)))	0,1	0
exact_match(firstnotitle(no_case(no_det(M_i,M_j))))	0,1	0
exact_match(last(M_i,M_j))	0,1	1
exact_match(last(no_case(M_i,M_j))	0,1	1
exact_match(rarest(M_i,M_j))	0,1	1
exact_match(rarest(no_case(M_i,M_j)))	0,1	1
exact_match(no_case(M_i,M_j))	0,1	0
exact_match(no_punct(M_i,M_j))	0,1	0
exact_match(no_case(no_punct(M_i,M_j)))	0,1	0
exact_match(no_det(M_i,M_j))	0,1	0
exact_match(no_case(no_det(M_i,M_j)))	0,1	0
exact_match(no_punct(no_det(M_i,M_j)))	0,1	0
exact_match(no_case(no_punct(no_det(M_i,M_j))))	0,1	0
MED_w(M_i,M_j)	continuous	1
MED_w(no_det(M_i,M_j))	continuous	1
MED_s(M_i,M_j)	continuous	4
MED_s(no_det(M_i,M_j))	continuous	4
MED_s(head(M_i,M_j)	continuous	0
MED_w(no_case(M_i,M_j))	continuous	1
MED_w(no_case(no_det(M_i,M_j)))	continuous	1
MED_s(no_case(M_i,M_j))	continuous	4
MED_s(no_case(no_det(M_i,M_j)))	continuous	4
MED_s(head(no_case(M_i,M_j)))	continuous	0
MED_w(no_punct(M_i,M_j))	continuous	1
MED_w(no_punct(no_det(M_i,M_j)))	continuous	1
MED_s(no_punct(M_i,M_j))	continuous	4
MED_s(no_punct(no_det(M_i,M_j)))	continuous	4
MED_w(no_case(no_punct(M_i,M_j)))	continuous	1
MED_w(no_case(no_punct(no_det(M_i,M_j))))	continuous	1
MED_s(no_case(no_punct(M_i,M_j)))	continuous	4

Table 3.8: Name-matching features and their values for ("New York", "York"), `no_web_parser+no_normalization` configuration shown in italic (continued).

Feature	Range	Example value
MED_s(no_case(no_punct(no_det(M_i,M_j))))	continuous	4
MED_w_anaph(M_i,M_j)	continuous	0.5
MED_w_anaph(no_det(M_i,M_j))	continuous	0.5
MED_s_anaph(M_i,M_j)	continuous	0.5
MED_s_anaph(no_det(M_i,M_j))	continuous	0.5
MED_s_anaph(head(M_i,M_j))	continuous	0
MED_w_anaph(no_case(M_i,M_j))	continuous	0.5
MED_w_anaph(no_case(no_det(M_i,M_j)))	continuous	0
MED_s_anaph(no_case(M_i,M_j))	continuous	0.5
MED_s_anaph(no_case(no_det(M_i,M_j)))	continuous	0.5
MED_s_anaph(head(no_case(M_i,M_j)))	continuous	0
MED_w_anaph(no_punct(M_i,M_j))	continuous	0.5
MED_w_anaph(no_punct(no_det(M_i,M_j)))	continuous	0.5
MED_s_anaph(no_punct(M_i,M_j))	continuous	0.5
MED_s_anaph(no_punct(no_det(M_i,M_j)))	continuous	0.5
MED_w_anaph(no_case(no_punct(M_i,M_j)))	continuous	0.5
MED_w_anaph(no_case(no_punct(no_det(M_i,M_j))))	continuous	0.5
MED_s_anaph(no_case(no_punct(M_i,M_j)))	continuous	0.5
MED_s_anaph(no_case(no_punct(no_det(M_i,M_j))))	continuous	0.5
MED_w_ante(M_i,M_j)	continuous	1
MED_w_ante(no_det(M_i,M_j))	continuous	1
MED_s_ante(M_i,M_j)	continuous	1
MED_s_ante(no_det(M_i,M_j))	continuous	1
MED_s_ante(head(M_i,M_j))	continuous	0
MED_w_ante(no_case(M_i,M_j))	continuous	1
MED_w_ante(no_case(no_det(M_i,M_j)))	continuous	1
MED_s_ante(no_case(M_i,M_j))	continuous	1
MED_s_ante(no_case(no_det(M_i,M_j)))	continuous	1
MED_s_ante(head(no_case(M_i,M_j)))	continuous	0
MED_w_ante(no_punct(M_i,M_j))	continuous	1
MED_w_ante(no_punct(no_det(M_i,M_j)))	continuous	1
MED_s_ante(no_punct(M_i,M_j))	continuous	1
MED_s_ante(no_punct(no_det(M_i,M_j)))	continuous	1
MED_w_ante(no_case(no_punct(M_i,M_j)))	continuous	1
MED_w_ante(no_case(no_punct(no_det(M_i,M_j))))	continuous	1
MED_s_ante(no_case(no_punct(M_i,M_j)))	continuous	1
MED_s_ante(no_case(no_punct(no_det(M_i,M_j))))	continuous	1
abbrev1(M_i,M_j)	0,1	0

Table 3.8: Name-matching features and their values for ("New York", "York"), `no_web_parser+no_normalization` configuration shown in italic (continued).

Feature	Range	Example value
abbrev1(no_case(M_i,M_j))	0,1	0
abbrev2(M_i,M_j)	0,1	0
abbrev2(no_case(M_i,M_j))	0,1	0
abbrev3(M_i,M_j)	0,1	1
abbrev3(no_case(M_i,M_j))	0,1	1
abbrev3(no_punct(M_i,M_j))	0,1	1
abbrev3(no_case(no_punct(M_i,M_j)))	0,1	1
abbrev4(M_i,M_j)	0,1	0
abbrev4(no_case(M_i,M_j))	0,1	0
abbrev4(no_punct(M_i,M_j))	0,1	0
abbrev4(no_case(no_punct(M_i,M_j)))	0,1	
rarest+contain(M_i,M_j)	0,1	1
rarest+contain(no_case(M_i,M_j))	0,1	1
rarest+contain(M_j,M_i)	0,1	1
rarest+contain(no_case(M_j,M_i))	0,1	1
matched_part_w(M_i, M_j)	continuous	1
matched_part_w(no_case(M_i, M_j))	continuous	1
matched_part_w(no_punct(M_i, M_j))	continuous	1
matched_part_w(no_det(M_i, M_j))	continuous	1
matched_part_w(no_case(no_det(M_i, M_j)))	continuous	1
matched_part_w(no_case(no_punct(M_i, M_j)))	continuous	1
matched_part_w(no_det(no_punct(M_i, M_j)))	continuous	1
matched_part_w(no_case(no_det(no_punct(M_i, M_j)))	continuous	1
matched_part_s(M_i, M_j)	continuous	4
matched_part_s(no_case(M_i, M_j))	continuous	4
matched_part_s(no_punct(M_i, M_j))	continuous	4
matched_part_s(no_det(M_i, M_j))	continuous	4
matched_part_s(no_case(no_det(M_i, M_j)))	continuous	4
matched_part_s(no_case(no_punct(M_i, M_j)))	continuous	4
matched_part_s(no_det(no_punct(M_i, M_j)))	continuous	4
matched_part_s(no_case(no_det(no_punct(M_i, M_j)))	continuous	4

Coreference-level features describe the similarity (match) between two NP strings. We have implemented a family of coreference-level features, aiming at comparing them and finding the best setting. Each feature can be represented by a triple (*normalization*, *substring selection*, *matching*). As was noted above, we apply the same substring selection algorithm to both an anaphor and an antecedent.

Not all the triples are useful. For example, combining `matched_part` algorithm with any substring selection would always produce the same values as `exact_match` with this substring. This observation reduces the total number of features dramatically. Overall we have 134 features, listed in Table 3.8: 26 markable-level features and 108 coreference-level features, represented by matching triples.

Configurations. In our experiments we want to test such hypotheses as, for example, "Approximate matching yields better results than exact matching" or "Case normalizing improves performance". In other words, we want to compare the system's performance in the cases when several features are activated and several ignored. We call these groups of activated features *configurations*. Below we describe the configurations we used in our experiments:

all: all 134 features

baseline1: exact matching for full names, no normalization

baseline2: all baseline1 features, `(no_normalization,head, exact_match)`

MED+head: all baseline1 features, minimum edit distance (MED) based triples: `(__,__, approximate_match)`

MED-head: all baseline1 features, `(__,no_substring_selection,approximate_match)` triples

MED_w-head: all baseline1 features, `(__,no_substring_selection, approximate_match)` triples, minimum edit distance measured in words (`MED_w`, `MED_w_anaph`, `MED_w_ante` in Table 3.5)

MED_s-head: all baseline1 features, `(__,no_substring_selection, approximate_match)` triples, minimum edit distance measured in symbols (`MED_s`, `MED_s_anaph`, `MED_s_ante` in Table 3.5)

MED_bare-head: all baseline1 features, `(__,no_substring_selection, approximate_match)` triples, minimum edit distance not normalized (`MED_s`, `MED_w` in Table 3.5)

MED_ante-head: all baseline1 features, (__,no_substring_selection, approximate_match) triples, minimum edit distance normalized by the length of antecedent(MED_s_ante, MED_w_ante in Table 3.5)

MED_anaph-head: all baseline1 features, (__,no_substring_selection, approximate_match) triples, minimum edit distance normalized by the length of anaphor(MED_s_anaph, MED_w_anaph in Table 3.5)

last: all baseline1 features, (__,no_substring_selection, exact_match) and (__,last, exact_match) triples

first: all baseline1 features, (__,no_substring_selection, exact_match) and (__,first, exact_match) triples

rarest: all baseline1 features, (__,no_substring_selection, exact_match), (__,rarest, contain and (__,rarest, exact_match) triples

no_MED: all features except (__,__, approximate_match) triples

no_abbrev: all features except (__,__, abbreviation) triples

no_web: all baseline1 features, triples that do not require Internet counts (i.e., no rarest-based features used)

no_web_parser: all baseline1 features, triples that require neither Internet counts nor parsing (i.e., all types of matching except abbreviation for full NP strings and their first and last substrings)

Each configuration can be used with different normalization strategies. Features for the no_web_parser+no_normalization configuration are shown in Table 3.8 in italic.

3.5 Experiments

Below we describe two experiment conducted to evaluate our matching algorithms. We rely on two machine learners, Ripper (Cohen, 1995) and SVM^{light} (Joachims, 1999) to classify markable pairs as $\pm matching$. SVM^{light}is a machine learner used throughout this thesis. Section 2.4 describes our motivation for preferring this learning algorithm. We perform a 10-fold cross-validation in our Experiment 1, which is a too time-consuming setting for relatively slow support vector learning. This motivates our decision to use the much faster Ripper system.

For our experiments we consider only proper names, as identified by the C&C NE-tagger (see Section 2.4 for a brief description and (Curran and Clark, 2003b) for more details). An oracle system, that correctly resolved

all NE markables, would have 30.1% recall and 98.0% precision on the MUC-7 test data and 26.3% recall and 99.2% precision on the MUC-7 dry-run data.

This evaluation set-up poses two challenges for an automatic name-matching approach. First, our data contain mentions of very similar, but still different entities, for example, names of relatives. Without any prior knowledge, it is very difficult to determine that "Bill Clinton" and "Hillary Clinton" are names of two different persons, unlike "Bill Gates" and "William Gates". This is a true matching problem, showing the limitations of a knowledge-poor approach.

Second, coreference is a more complex phenomenon than matching, involving many different factors. For example, the same name can be used metonymically to denote different entities. Thus, one of the MUC-7 documents contains 3 coreference chains for "McDonald's" (for the company itself, a particular eating place, and a general concept of such a place). In another document, "Beijing" is coreferent with "China", both of them meaning "the Government". Therefore, using coreference data for name matching, we inevitably introduce noise into the training data and error into the test data. This is not a true matching problem, but rather a drawback of the experimental design. Unfortunately, we are not aware of any Natural Language dataset annotated specifically for name matching.

3.5.1 Experiment 1: Instance-based Evaluation

In this experiment we use only the training data (30 "dry-run" texts) and the Ripper machine learner. The experiment has been organized as follows. For each of our 10 cross-validation runs, we reserve 3 texts for testing. The remaining texts are first used to optimize Ripper's S parameter (degree of hypothesis simplification): we perform 3-fold cross validation on these texts, varying S from 1 to 10, in order to find the best setting. This is done to avoid overfitting. We do not optimize any other parameters. Finally, we train Ripper on all the 27 texts with the best S value and test on the reserved 3 texts.

In this experiment we measure the performance in the standard way (precision is the ratio of correctly suggested anaphoric links over all the anaphoric links suggested by Ripper and recall is the ratio of correctly suggested anaphoric links over the total of pairs). So, this experiment follows a database setting: each entry (name) is compared to all the preceding ones, no notion of chains or sets of co-referring names is supported.

The results of our first evaluation experiment are shown in Table 3.9. All the configurations perform significantly ($p < 0.01$, two-tailed t-test) better than the first baseline. The configurations, yielding significantly ($p < 0.05$) better results than the second baseline are marked with †. All the counts are compared to the `no normalization` baseline cases. Recall that our baselines do not make use of any form of normalization. We show the performance of

the normalized baseline settings in parentheses.

Overall, the worst performing configuration is the first baseline, i.e. only exact matching of full strings without any normalization. The best performing configuration ($F = 84.6\%$) is a combination of all the matching functions but `abbreviations` with the `no_determiner` normalizing function.

3.5.2 Experiment 2: MUC-style Set-based Evaluation

Experiment 1 has assessed the quality of our name-matching features in a simulated database scenario: we measured performance on pairs of markables. In this experiment we shift to a set-based evaluation scheme, adopted by most studies on coreference.

We have used the dry-run data for training and the 3 MUC-7 "training" documents for testing. The evaluation figures have been obtained with the MUC scoring program (Vilain et al., 1995). The scorer does not compare individual pairs, but whole coreference chains for estimating the system's performance (see Section 2.2). We have implemented a basic coreference resolution algorithm, as described in Section 2.3.

The results of the set-based evaluation for the most important configurations are shown in Tables 3.10 and 3.11. Both our baselines have very low recall (65.9% for *baseline1*). All the other configurations, except `first` and `last` in the Ripper case, have a similar precision, but significantly ($p < 0.05$, χ^2-test) higher recall[4].

The experimental results reported so far only address coreference resolution for named entities. They show that a majority of proper names can be resolved with simple and shallow matching techniques. Other types of anaphors are more difficult. We have assessed the applicability of name-matching for full-scale coreference resolution by re-training the system on the whole corpus (including common noun phrases and pronouns).

We have trained the SVM^{light} classifier with all our 134 features and evaluated it on the 20 "formal test" documents. It achieves a recall level of 52.2% and a precision level on 61.2%. These figures suggest that name-matching is a good starting point for building a full-scale coreference resolution engine, but deeper knowledge should be added to account for non-NE anaphors and thus obtain a better classification.

3.5.3 Discussion

As our evaluation experiments show, sophisticated matching algorithms clearly outperform the baselines. By choosing the right configuration, one can boost the system's performance dramatically — the difference in the performance between the best and the worst configuration in our first experiment was 26%

[4]We cannot compare the corresponding F-scores, as the χ^2-test is not applicable.

Configurations	no normalization	no_case	no_punctuation	no_determiner	full normalization	all normalizations together
Fast and Language Independent						
baseline1	58.7	(63.9)	(63.2)	(64.1)	(63.8)	(63.9)
first	65.5	74.1	75.0	74.6	74.4	75.0
last	69.9	69.3	71.4	71.6	71.3	70.2
no_web_parser	79.8	†82.0	81.3	79.4	†82.4	80.7
MED-head	76.4	†81.7	†82.7	†82.5	†82.1	81.4
MED_w-head	74.9	81.3	†81.9	†82.4	†82.7	81.0
MED_s-head	70.7	73.8	75.2	77.3	75.3	78.0
MED_bare-head	71.2	79.3	76.6	75.3	77.7	78.3
MED_ante-head	75.9	81.3	79.7	81.6	†81.3	80.6
MED_anaph-head	72.9	73.9	74.2	76.2	79.0	75.8
Using Parsing						
baseline2	75.9	(†81.4)	(†80.7)	(†80.9)	(†81.3)	(†81.2)
MED+head	†81.9	†84.3	†83.3	†83.2	†82.6	†83.8
no_MED	†83.5	†83.4	†84.6	†83.7	†84.5	†83.6
no_abbrev	†83.2	†83.5	†84.2	†84.6	†84.5	†83.9
no_web	81.6	†82.3	†83.4	†83.7	†83.1	†80.7
Using Web						
rarest	79.0	81.4	80.3	†82.0	†81.9	†81.3
Using Parsing and Web						
all	†82.4	†83.0	†82.3	†83.9	†83.5	†82.0

Table 3.9: Comparing different configurations: the classifier's performance (F-measure) in the 10-fold cross-validation on the MUC-7 dry-run data. All the configurations performed significantly ($p < 0.01$, two-tailed t-test) better than the first baseline. The configurations, yielding significantly ($p < 0.05$) better results than the second baseline are marked with †.

Configurations	no normalization			all norm. together		
	Recall	Precision	F	Recall	Precision	F
baseline1	65.9	51.8	58.0	(73.9)	(53.7)	(62.2)
first	65.9	51.8	58.0	73.9	53.7	62.2
last	78.4	49.6	60.8	81.8	49.7	61.8
no_web_parser	80.7	51.1	62.6	81.8	50.3	62.3
MED-head	65.9	51.8	58.0	73.9	53.7	62.2
baseline2	65.9	51.8	58.0	(73.9)	(53.7)	(62.2)
MED+head	65.9	51.8	58.0	73.9	53.7	62.2
no_web	84.1	52.5	64.6	87.5	52.4	65.5
rarest	81.8	52.6	64.0	81.8	52.6	64.0
all	84.1	53.2	65.2	87.5	52.4	65.5

Table 3.10: Comparing different configurations: the classifier's performance in set evaluation on the validation data, the SVM^{light} learner.

Configurations	no normalization			all norm. together		
	Recall	Precision	F	Recall	Precision	F
baseline1	65.9	51.8	58.0	(73.9)	(53.7)	(62.2)
first	65.9	51.8	58.0	80.7	52.6	63.7
last	78.4	49.6	60.8	84.1	48.7	61.7
no_web_parser	86.4	49.0	62.6	88.6	51.7	65.3
MED-head	86.4	52.4	65.2	86.4	53.5	66.1
baseline2	72.7	50.4	59.5	(78.4)	(51.1)	(61.9)
MED+head	84.1	49.3	62.2	86.4	50.7	63.9
no_web	89.8	49.4	63.7	85.2	51.7	64.4
rarest	83.0	48.0	60.8	73.9	53.7	62.2
all	92.0	52.9	67.2	90.9	49.7	64.3

Table 3.11: Comparing different configurations: the classifier's performance in set evaluation on the validation data, the Ripper learner.

(63% error reduction). However, tuning these algorithms to achieve the best performance is not a trivial task.

The substring selection proves to be useful: both `head` and `MED+head` configurations perform reasonably well. The Internet-based substring selection (`rarest`) is only slightly worse for the training data (cross-validation results) and even better for the test data. Similar techniques, however, can bring only moderate advantage over the (first) baseline: our `first` and `last` configurations do not yield reliable performance. So, if we want, for example, to build coreference resolution systems for other languages, where parsing resources are less reliable or even non-existent, we should try another solution: either use several substrings at the same time (as in the `no_web_parser` case), or improve other parts of our matching algorithms, for example, use MED-related features.

Our sophisticated matching functions improve the system's performance to some extent, although abbreviations seem to be almost useless. The most important function is approximate matching. In fact, MED is such a powerful technique that all the additional improvements do not affect the performance significantly: consider the differences in F-measure for `MED-head` (with the `head` substring selection) and `MED+head` (without any substring selection).

As far as different MED-related features are concerned (cf. Table 3.5), it is usually better to normalize MED-counts by the length of the antecedent than by the length of the anaphor or not normalize at all, and the very best solution is to use both normalized and not normalized counts at the same time (`MED-head` and `MED-head`). Also, the distance measured in words (`MED_w`) in general works better than the distance measured in symbols (`MED_s`). However, most of these differences are not statistically significant.

Finally, we could not find a normalization function that would outperform all others in all cases. But our experiments show that it is worth using at least some normalization: `no normalization` results in a significant drop of the performance for almost all configurations. In the approximate matching configurations the type of the normalization function does not normally play an important role ($2 - 3\%$ difference in F-measure, except for the `no normalization` case). With `exact_match`, normalization becomes more important (up to 10% difference in the F-measure).

Clearly, the performance gets higher with more resources available. However, by using a variety of advanced matching techniques, one can obtain promising results even with a very shallow and fast approach, not relying on a parser or web counts. This result is especially important for future work: it shows that accurate name matching should be possible even for languages with scarce parsing resources and much smaller web coverage than English.

3.6 Summary

In this chapter we have addressed the problem of name matching for coreference resolution. We have seen that most state-of-the-art systems rely on very simple matching techniques, although much more advanced solutions have been proposed in the literature. We have given an overview of the relevant approaches in Section 3.1.

Section 3.2 has provided a brief discussion of the most typical name-matching problems, relevant for the coreference resolution task. In Section 3.3, we have carried out a systematic investigation of possible extensions to a naive name-matching algorithm, decomposing the name matching problem into three major sub-tasks:

- normalization: `no_case`, `no_punctuation`, and `no_determiner`;
- substring selection: `head`, `last`, `first`, and `rarest`;
- matching: `exact_match`, `approximate_match` (Minimum Edit Distance), `matched_part` (overlap), `abbreviation`;

We obtain a surface similarity measure between two markables, by specifying each of these sub-algorithms and combining them into the "matching triples", for example, `exact_match(first(no_det(`M_1, M_2`)))`.

After discarding trivial triples, we have come up with 134 lexicographic features, described in Section 3.4. We have trained the Ripper information gain-based rule induction system and the SVM^{light} Support Vector Machine-based learner with these features while considering only named entities as markables.

Our evaluation experiments in Section 3.5 show, that sophisticated matching algorithms clearly outperform the baselines. By choosing the right configuration, one can boost the system's performance dramatically, achieving up to 63% error reduction in the instance-based evaluation (22% in the MUC-style evaluation).

The evaluation figures suggest that surface similarity is a reliable indicator for coreference, but it still cannot account for all types of anaphoric links. In the next chapter we will look at how syntactic information may influence coreference resolution.

Chapter 4

Syntactic Knowledge

Earlier studies in Coreference Resolution (Hobbs 1978, among others), in particular, in pronominal anaphor, have exploited various syntactic properties of markables and their contexts. This resulted in a number of heuristics that can be encoded as hard constraints, filters, preferences, or features to guide a coreference resolution algorithm. Although none of the existing approaches relies solely on syntactic information, the latter constitutes an important part of all the algorithms for anaphora resolution.

Recent advances in parsing technology (Charniak, 2000; Collins, 1999) make syntactic information more and more valuable for any kind of natural language processing: with a performance level of around 90%, state-of-the-art parsers and taggers provide reliable and robust knowledge.

Syntactic information is especially important for intrasentential coreference, outweighing other factors in most cases. Consider the following example:

(21) Under the current agreement, survivors can bring suit in [the country]$_1$ the plane was flying to, [the home country]$_2$ of the airline, [a country]$_3$ where the airline has its major operation or [the country]$_4$ where the contract for transportation was written.

The noun phrases in brackets, NP_1, NP_2, NP_3, and NP_4, are lexicographically very similar and semantically compatible. Most salience features are not applicable in this case. Nevertheless, these markables obviously can not corefer, which can be predicted by taking syntactic information into account: each NP commands the following ones (see Section 4.6), all NPs are postmodified (see Section 4.5), and the third NP has a "non-anaphoric" determiner (see Section 4.3).

Around 30% of markables in our corpus have antecedents in the same sentence. For pronominal anaphors, this figure is much higher: 61% and 47% of pronouns have intrasentential antecedents in the training and validation corpus respectively. Tetreault (2001) shows that intrasentential coreference might be even more important for other domains: around 70% of pronouns in their New York Times-based corpus have antecedents in the same sentence.

On the intersentential level, syntactic analysis provides valuable information that can be combined with other features (see Chapter 8). For example, the `syntactic_agreement` constraints filter out undesired candidates and the `type_of_markable` feature may guide the whole resolution process, if separate sub-algorithms are used for different types of anaphors.

Syntactic information also provides a backbone for extracting other types of features. Thus, our semantic module (Chapter 5) compares the markables' `heads`, the `grammatical_roles` are used to compute salience ranking (Chapter 6), and all the NP-level syntactic features help to build the anaphoricity classifier (Chapter 7).

Most syntactic principles are traditionally formulated as hard constraints, for example, "Two NPs do not corefer if they disagree in number" or "Two NPs do corefer if they are parts of an apposition". Those constraints have usually been developed and tested on small sets of manually analyzed examples. Such methodology leads to two problems. First, some tricky examples may not follow the expected patterns: "they" and "the company" disagree in number, but still can corefer, because "the company" refers to a group entity. Second, preprocessing errors may also affect the accuracy: in our data, "they" and "Iraqis" disagree in number, because "Iraqis" is mis-tagged by the parser as "NNP", and, thus, considered singular. Several studies, in particular Mitkov (1997), have shown benefits and drawbacks of a constraint-based architecture. In our approach we use a feature encoding for capturing syntactic constraints.

In this chapter we investigate various kinds of syntactic information discussed in the literature (Section 4.1). We examine in Sections 4.2–4.9, to what extent particular kinds of syntactic knowledge may influence the distribution of anaphors/antecedents (for markable-level features) or coreference links (for coreference-level features). We then use our features to experimentally assess the importance of syntactic knowledge for intrasentential anaphora and for full-scale coreference resolution (Section 4.10).

4.1 Related Work

Numerous linguistic accounts exist for all the syntactic information discussed in this chapter. These theories are often very specific and, at the same time, complex and therefore we are not able to summarize all of them here. Each of

the following sections, however, points to some relevant studies. In this section we focus on more general syntax-based research on anaphora.

The syntactic conditions that influence the interpretation of anaphora have been one of the major topic in transformational grammar. This framework does not imply a full resolution procedure, but rather aims at at identifying possible referential relations that can hold between two markables (Reinhart, 1983):

(22) A. Obligatory (stipulated) coreference:
[Zelda]$_1$ bores [herself]$_{2,ante=1}$.

B. Obligatory (stipulated) non-coreference:
[Zelda]$_1$ bores [her]$_{2,ante\neq1}$.
[She]$_1$ adores [Zelda]$_{2,ante\neq1}$'s teachers.

C. Optional (free) coreference:
[Zelda]$_1$ adores [her]$_2$ teachers.
Those who know [her]$_1$ adore [Zelda]$_2$.

The third person pronoun "herself" in (22A) can only be resolved to "Zelda". In (22B), on the contrary, both pronouns cannot corefer with "Zelda". Both readings are plausible for (22C).

To account for these cases, the transformational framework relies on such notions as *command*, especially *c-command*, and *domain*, introduced by Langacker (1969) and developed in later studies. We discuss these ideas in detail in Section 4.6 below.

The study of Hobbs (1978) addresses the problem of anaphora resolution from an analytic point of view and, unlike the above-mentioned approaches suggests a resolution procedure. Hobbs (1978) uses syntactic knowledge and NP gender information to develop an algorithm for (pronominal) coreference resolution. Although very different from Langacker's (1969) approach at first glance, this study implicitly relies on the command rules proposed by the transformational grammarians by imposing left-to-right breadth-first search order.

Later studies, for example, (Reinhart, 1983; Pinkal, 1991) have criticized the underlying theoretical assumption of the whole transformational framework, arguing that co- and contra-indexing rules are a pure syntactic mechanism which cannot by itself account for specifying anaphoric relations. Instead, they propose to incorporate coindexing information into semantic processing, for example, in the DRT-style (Latecki and Pinkal, 1990).

To summarize, earlier papers on generative grammar introduce several related notions of "command", formalizing contra-indexing constraints. Later studies, however, show that these constraints are not the only and probably not even the main factors in the interpretation of anaphors.

State-of-the-art algorithms, both for full-scale and pronominal coreference resolution, combine syntactic knowledge (command constraints and simple versions of agreement-based filters) with more sophisticated semantic and salience parameters. Unfortunately, the evaluation study of Tetreault (2001) shows that these additional information sources might be not as beneficial as they seem: Hobbs's (1978) naive algorithm still outperforms most salience-based approaches.

4.2 Types of Markables

Intuitively, different kinds of anaphors require very different resolution procedures. This intuition is supported by linguistic theory (Webber, 1979). For example, pronouns usually have their antecedents in a few preceding sentences, whereas coreferring named entities can appear very far from each other. Therefore it is important to have a classification of markables (in particular, anaphors) into "types", reflecting common resolution strategies.

Most state-of-the-art approaches to the full-scale coreference resolution task apply a uniform resolution procedure to all the anaphors[1]. For example, Soon et al. (2001) learn and test their classifiers on the full set of NPs. However, even those approaches usually encode "type of a markable" in some way. Thus, Soon et al. (2001) rely such features as "anaphor is a pronoun" or "both the anaphor and the antecedent are proper names" (see Section 2.3).

Theoretical studies both in linguistics (Webber, 1979; Gundel et al., 1993) and psychology (Garrod et al., 1994), on the contrary, clearly suggest different processing mechanisms for various types of anaphors.

Several algorithms have been proposed to handle specifically just one kind of anaphors: pronouns (an overview of modern pronoun resolution algorithms can be found in (Mitkov, 1999)), definite descriptions (Vieira and Poesio, 2000; Gasperin et al., 2004), or named entities (Bontcheva et al., 2002; McCallum and Wellner, 2003). The diversity of these algorithms, both in their structure and their knowledge sources, suggests, in accordance with the above-mentioned theoretical research, that a full-scale coreference resolution engine might benefit from distinguishing between markables of different types.

We following theoretical studies and identify different `types_of_markables`: PRONOUN, NAMED ENTITY, DEFINITE NP, and OTHER. PRONOUNs are further subdivided into PERSONAL, POSSESSIVE, and REFLEXIVE. DEFINITE NPs are subdivided into The-NPs (i.e., NPs with the definite article "the") and DT-NPs (i.e., NPs with other definite determiners, see

[1]Poesio and Alexandrov-Kabadjov (2004) propose an architecture, allowing different processing strategies for different types of anaphors. Their system, GUITAR, supports different subclassifiers for various types of anaphors, in particular, pronouns and definite descriptions. These subclassifiers or even different resolution sub-algorithms can then be combined in a uniform framework.

Type	Anaphors				Antecedents			
	+anaphor		−anaphor		+ante		−ante	
DEFNP	437	40%	660	60%	364	33%	733	67%
The-NP	380	42%	531	58%	307	34%	604	66%
DT-NP	57	31%	129	69%	57	31%	129	69%
NE	578	34%	1138	66%	627	37%	1089	63%
PRONOUN	363	87%	55	13%	270	65%	148	35%
Personal	242	84%	45	16%	176	61%	111	39%
Possessive	114	92%	10	8%	89	72%	35	28%
Reflexive	7	100%	0	0%	5	71%	2	29%
OTHER	229	13%	1568	87%	345	19%	1452	81%

Table 4.1: Distribution of anaphors vs. non-anaphors (left) and antecedents vs. non-antecedents (right) for different types of markables in the training data (30 MUC-7 "dry-run" documents).

"DET_ana" determiners in Section 4.3 below). This information can be computed straightforwardly from the output of a parser and NE-recognizer. If a markable is both a named entity and an NP at the same time, it is considered NE (and not Definite or Other NP).

We have computed the distribution of anaphors vs. non-anaphors and antecedents vs. non-antecedents for different markables' types in our training set (Table 4.1). We have conducted a χ^2-test in order to see, whether the `type_of_markable` variable interacts with anaphoricity (antecedenthood). The χ^2-test suggests a clear interaction ($p < 0.01$). This is in accordance with the theoretical claim that different types of markables behave differently with respect to coreference.

4.3 Determiner

Determiners play an important role in the interpretation of NP-anaphors. Linguistic theories (for example, (Prince, 1981), see Section 7.1) generally assume that indefinite NPs are used to introduce objects, whereas definite NPs refer to already known entities: subsequent mentions ("A house... the house..."), indirect anaphors ("A house... the door..."), objects from the surrounding extra-linguistic context ("This house is protected by CCTV"), or well-known unique entities ("The White House"). Some determiners, such as "no" or "any" typically introduce non-referring markables (Karttunen, 1976).

Recent corpus-based studies, however, confirm this theoretical view only partially, highlighting several problematic cases. Vieira and Poesio (2000) show that more than 50% of definites in their corpus are not anaphoric. Nis-

sim (2002) suggests various factors influencing the surface structure of indirect anaphors, in particular, the distribution of possessive forms vs. definite articles. The MUC-7 data contain a few examples of anaphoric indefinite NPs, in their majority parts of appositions and copula constructions.

Using the parser's output, we compute the `determiner` value for a given markable as follows:

1. If the markable is a named entity or a pronoun, the determiner function is not applicable ("N/A" value). These markables are excluded from the analysis presented in this section.

2. If the markable starts with a determiner ("DT" tag), we retrieve and lower-case its surface form.

3. If the markable contains an embedded possessive NP ($[\texttt{.+ ['s}]_{POSS}]_{NP}$) or is premodified with a possessive pronoun, we classify it as "0_POSS"

4. If none of these conditions holds, we classify the markable as "0". This is a very heterogeneous class, containing bare nominals, incorrectly parsed NPs and one-word markables extracted from auxiliary parts of MUC documents (see Section 2.5).

We have analyzed the distribution of anaphoric vs. non-anaphoric markables for different determiners in the MUC-7 dryrun corpus in order to see, how the surface form of the determiner may affect the anaphoricity status of an NP. These data are shown in table 4.2. In total, 764 of 3123 noun phrases (24.5%) in the training corpus are anaphoric. If a markable is used with "the", "this", or "these", it is more likely (χ^2-test, $p < 0.05$) to be an anaphor, whereas if a markable is used with "no", "a(n)", or with no article, it's more likely (χ^2-test, $p < 0.05$) to be not anaphoric.

Our corpus-based estimations lead us to distinction of the following three classes of determiners: DET_ana ("the", "this", "these"), DET_nonana ("0", "a(n)", "no"), and DET_other ("0_POSS", "all", "that", "those", "some", "another", "any", "each", "both", "half", "every"). We will refer to the DET_ana class as "definite determiners" throughout the thesis. It must be noted that our classifications, obtained with pure data-driven techniques, are generally in accordance with linguistic research on anaphoricity.

The left part of Table 4.3 presents the distribution of anaphoric vs. non-anaphoric markables for different classes of determiners in our training data. Our validation corpus follows the same pattern: the distribution of anaphoric vs. non-anaphoric markables depends on the extracted determiner type (χ^2-test, $p < 0.01$). This shows that the classifications are linguistically relevant and do not simply reflect peculiarities of the training corpus.

Finally, we have run a similar experiment to investigate the distribution of antecedents vs. non-antecedents for markables with different determiners

Type	+anaphor		−anaphor		+antecedent		−antecedent	
0	150	11%	1206	89%	235	17%	1121	83%
the	464	44%	579	56%	387	37%	656	63%
a(n)	75	19%	312	81%	103	27%	284	73%
0_POSS	41	21%	155	79%	55	28%	141	72%
this	12	41%	17	59%	9	31%	20	69%
no	0	0%	20	100%	0	0%	20	100%
all	3	16%	16	84%	4	21%	15	79%
that	7	41%	10	59%	4	24%	13	76%
those	2	18%	9	82%	4	36%	7	64%
some	0	0%	11	100%	1	9%	10	91%
these	8	89%	1	11%	4	44%	5	56%
another	1	11%	8	89%	1	11%	8	89%
any	0	0%	6	100%	0	0%	6	100%
each	0	0%	5	100%	1	20%	4	80%
both	1	33%	2	67%	0	0%	3	100%
half	0	0%	1	100%	1	100%	0	0%
every	0	0%	1	100%	0	0%	1	100%

Table 4.2: Distribution of anaphors vs. non-anaphors (left part) and antecedents vs. non-antecedents (right part) for markables with different determiners in the training data (30 MUC-7 "dry-run" documents), only full NPs are considered.

Type	+anaphor	−anaphor	Type	+ante	−ante
DET_ana	484	597	DET_ante	387	656
DET_nonana	225	1538	DET_nonante	235	1141
DET_other	55	224	DET_other	187	517

Table 4.3: Distribution of anaphors vs. non-anaphors (left part) and antecedents vs. non-antecedents (right part) for different determiner classes in the training data (30 MUC-7 "dry-run" documents).

in our corpus (right part of Table 4.3). Analyzing the training data, we have come up with the following classes: DET_ante ("the"), DET_nonante ("no", "0"), and DET_other (all the other determiners). These classes interact (χ^2-test, $p < 0.01$) with the antecedent vs. non-antecedent distribution in both the training and the validation data.

4.4 Head

One of the most important characteristics of a markable is its head word,, as it is used to compute many other parameters (for example, number, gender, some of the matching functions, etc). Therefore, we need an accurate procedure for determining markables' heads. Later in this section we also discuss the possibility of incorporating the head feature directly into a coreference resolution algorithm.

We analyze raw text data, provided in the MUC-7 distribution, with Charniak's (2000) parser (see Section 2.4 for a very brief overview). The heads of our markables are extracted from their parse trees with the following procedure (the markables are shown in square brackets and their heads in boldface):

- For a named entity, we simply consider its last noun word to be the head: [Trish **Neusch**], [NYTimes News **Service**] clients.
- Pronouns are their own heads: [**you**], [**your**].
- For full NPs, we use the algorithm proposed by Collins (1999) to compute the head.
- For possessive NPs, however, the algorithm of Collins (1999) makes an undesired prediction, considering "'s" to be the head: [the nation **'s**] surplus plutonium. In these cases, we remove the "'s" item from the parse and re-apply the same algorithm: [the **nation** 's] surplus plutonium
- In one special case, we identify two heads: an *NP-head* and an *NE-head*. Consider the following example (the parse tree is shown in Figure 4.1):

  [$_{NP}$ The Microsoft chairman [$_{NE}$ Bill Gates]]

  According to our procedure for extracting markables (see Section 2.5), this construction is one markable, which is both an NP and an NE. Strictly speaking, the syntactic head of the above NP is "Gates". However, this word does not give us a lot of information: for example, its `semantic_class` can be computed only on a very coarse level (PERSON) and its automatically extracted agreement features, in particular, `number`, although correct in this case, might potentially be misleading. Therefore we might want to consider the word "chairman" as the head

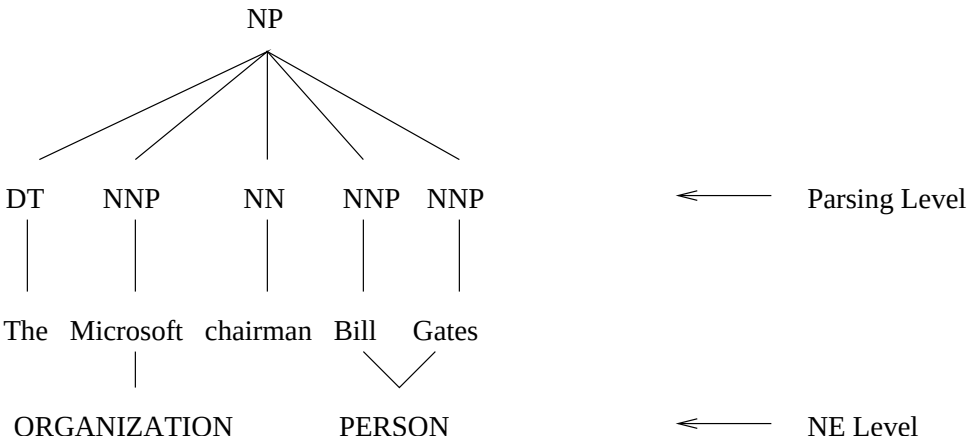


Figure 4.1: Parse tree for an NP-NE markable.

in this case. Unfortunately, this solution is not perfect either, as it now becomes very difficult to establish a coreference link between "the Microsoft chairman Bill Gates" and "Mr. Gates". To deal with this problem, we have introduced two heads for such markables: the `NP-head` ("chairman") encodes the semantically most important word and is used to compare the markable with another markables that are not named entities, whereas the `NE-head` ("Gates") represent the syntactic head and is used for name-matching with other named entities. The `NE-head` is the last noun of the NE part of the markable. To compute the `NP-head`, we take the whole markable and remove its NE-part (resulting in the "the Microsoft chairman" string in our example). We then apply the procedure of (Collins, 1999) to this truncated NP, provided it contains at least one noun.

- Another special case is coordination. There is no agreement among linguists on how to define the head of a coordinate construction (see (Kruijff, 2002) for an overview of relevant research).

  None of the commonly used solutions is fully appropriate in our case. We want, for example, all the following markables to have different heads to prevent incorrect matching: "Rear Admiral Husband E. Kimmel", "Maj. Gen. Walter C. Short", "Kimmel and Short", and "Kimmel or Short". That means that we want a head of a coordinate construction to contain the information about each conjunct and the conjunction itself. Following Hudson (1990), we introduce "external heads" for coordinated NPs, merging the conjunction and the heads of each conjunct: [[the two crew **members**] and [three **people**] on the ground]$_{head=\textbf{AND\_members\_people}}$ (see Figure 4.2 for the parse tree).

We have conducted two experiments similar to those described in Section

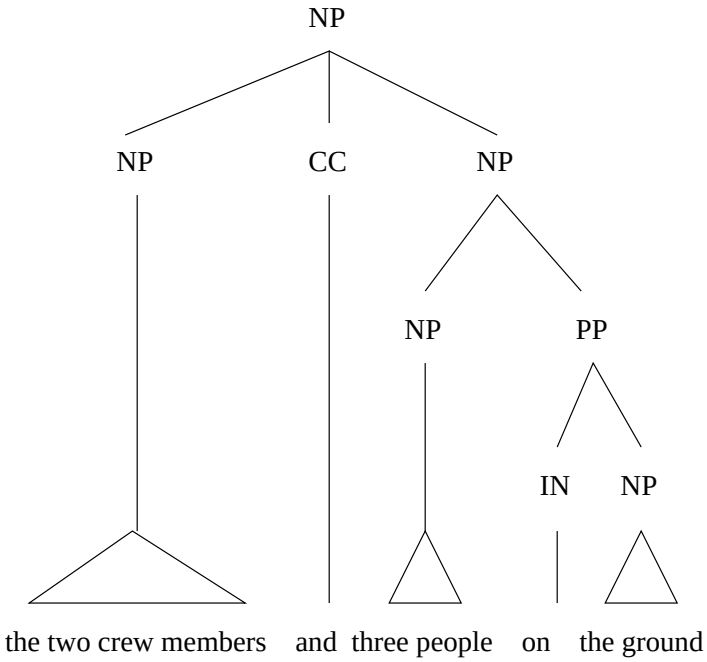


Figure 4.2: Parse tree for a coordination structure.

4.3 above, in order to analyze the interaction between the surface form of a markable's head and its anaphoricity properties.

Based on the training data we classify all the possible heads into (a) HEAD_anaphoric, HEAD_nonanaphoric, and HEAD_other or (b) HEAD_antecedent, HEAD_nonantecedent, and HEAD_other with the same procedure we have used for the determiners. Table 4.4 shows some examples. [2].

Unlike in the determiner case, these classifications, however, do not seem intuitively plausible. Thus, HEAD_anaphoric and HEAD_antecedent classes consist mainly of pronouns, proper names, and domain-specific or even document-specific common names. As for pronouns, we can identify them more reliably by checking their part-of-speech tags. Nouns on the lists only encode the peculiarities of the training data, reflecting the main topics of the documents or the whole domain.[3]

HEAD_nonanaphoric and HEAD_nonantecedent classes seem to be a bit more useful. Notice that they contain several common parsing mistakes ("---", ")"), auxiliary words from the header of the documents ("c.1996", "times"), and heads of numeric constructions ("percent", "miles"). Identifying these kinds of markables, especially parsing mistakes and auxiliary words, can be

[2]Most words belong to the HEAD_other class in both cases and are not shown in the Table.

[3]Training and test texts in MUC-7 are taken from different domains (air-crashes vs. politics/business).

Class	Surface form of the head
HEAD_anaphoric	they, our, them, we, you, md-88, navy, valujet, twa, faa, crash, airline, component, jet, 05-14-96, miami, chamberlin, kubeck, today, then
HEAD_nonanaphoric	art, use, ---, times,), miles, way, there, years, percent, operations, c.1996, 1994
HEAD_antecedent	it, his, me, him, he, i, we, captain, craft, mechanic, airline, 04-12, 05-14, china, jessica, kubeck
HEAD_nonantecedent	feet, art, way,), ---, there, operations, percent, miles, pieces, use, times, c.1996

Table 4.4: Examples of `heads` w.r.t. their anaphoricity distribution, only the training data (30 MUC-7 "dry-run" documents) used.

beneficial in a real-world system, where all the components are error-prone.

Tables 4.5 presents the distribution of anaphoric vs. non-anaphoric and antecedent vs. non-antecedent markables for different classes of heads. The numbers in parentheses show the same distribution after discarding all the pronouns.

A χ^2-test suggests that the head's class generally affects the distribution of anaphoricity for both the training and validation data. When we remove pronouns, however, we still get a strong effect ($p < 0.01$) for the training data, but no significant difference for the validation data, taking into account just HEAD_anaphoric (HEAD_antecedent) and HEAD_other classes. Comparing HEAD_nonanaphoric (HEAD_nonantecedent) against HEAD_other, we find a significant interaction on the validation data as well. A log-linear test clearly suggests interaction ($p < 0.001$) between all three parameters (`head_type`, `anaphoricity`, and `corpus_type` (validation, train)).

In sum, analyzing both word lists (Table 4.4) and the resulting distributions (Table 4.5), we conclude that it might be not beneficial to include the `head` feature into our algorithm directly: the coreference properties of head words are highly influenced by the peculiarities of the training material used, and, thus, such knowledge would only increase overfitting. The only potentially useful information comes from negative classes, HEAD_nonanaphoric and HEAD_nonantecedent, but even these distributions interact with corpus type.

Anaphors		
Type	+anaphor	−anaphor
HEAD_ana	861 (508)	240 (194)
HEAD_nonana	6	204
HEAD_other	740 (730)	2977 (2968)
Antecedents		
Type	+antecedent	−antecedent
HEAD_ante	796 (561)	291 (192)
HEAD_nonante	1	182
HEAD_other	809 (774)	2949 (2900)

Table 4.5: Distribution of anaphors vs. non-anaphors (left part) and antecedents vs. non-antecedents (right part) for different head classes in the training data (30 MUC-7 "dry-run" documents).

4.5 Internal Structure of Markables

Most coreference resolution systems take into account relations *between* markables. It might, however, be useful to look into their *internal* structure as well. For example, Poesio et al. (1997) have shown that different kinds of pre- and post-modification are important for the resolution of definite NPs.

We consider three syntactic constructions in this thesis: coordination, premodification, and post-modification.

Coordinations. Coordination is identified via a simple regular expression matcher:

`[NP|NN|NNP|NNS|NNPS|,]+ CC [NP|NN|NNP|NNS|NNPS]`.

We have several reasons to pay special attention to this particular construction. First, coordination of more than two NPs has a similar structure to apposition:

(23) The decision to go ahead was made Thursday night after [Symington]$_1$, [ASU officials]$_2$, [Host Committee members]$_3$ and [Neil Austrian]$_4$, [president of the NFL]$_{5,ante=4}$, watched eight landings by the helicopter.

The apposition feature is a very strong indicator for coreference, and it is therefore important to accurately distinguish between these two constructions at least in simpler sentences. Charniak's (2000) parser, used throughout this thesis, cannot reliably analyze complex cases of apposition and coordination. We will see in Section 8.2 how these errors affect the system's performance.

Second, several properties of a coordinate construction do not correspond to the properties of its parts. For example, the number (See Section 4.9) of

the whole construction is plural, even if each of its parts is singular:

(24) [[Symington]$_2$ and [Austrian]$_3$]$_1$ .. [They]$_{4,ante=1(\neq 2,\neq 3)}$

Coordination is problematic for ranking-based approaches, for example, Centering theory (see Section 6.5), as several entities share the same position in the (accessibility) hierarchy:

(25) [[Symington]$_2$ and [Austrian]$_3$]$_1$ .. [He]$_{4,ante\neq 2,\neq 3)}$

An accurate treatment of such a construction within a ranking-based framework would require introducing several partial orderings, complicating existing anaphora resolution algorithms.

Pre-modification. It can be argued that at least some pre-nominal modifiers are clear indicators for or against coreference. For example, the noun phrase "the same story" is a very likely anaphor, whereas "a different story" is more likely to be a discourse new description. We identify pre-modified descriptions by checking whether a markable contains any words between the determiner and the head: [a highly radioactive element].

Post-modification. Post-modification is a syntactic construction, where the head is not the last word of the markable. Post-modification can be either **restrictive** or **non-restrictive** (Vieira and Poesio, 2000). Restrictively post-modified NPs normally introduce new entities, and, thus, are claimed to be strong indicators for non-coreference. We consider the following patterns of restrictive post-modification in our thesis:
$[..]_{NP}\ [..]_{SBARQ}$
$[..]_{NP}\ [..]_{SBAR}$
$[..]_{NP}\ [..]_{S}$
$[..]_{NP}\ [..]_{VP}$
$[..]_{NP}\ [..]_{PP}$
$[..]_{NP}\ [..]_{WHPP}$

Tables 4.6 shows the distributions of anaphors vs. non-anaphors and antecedents vs. non-antecedents for coordination and pre/postmodification. They suggest interaction (χ^2-test, $p < 0.01$) between the coreference properties of a markable and its syntactic structure: more complex NPs seldom participate in coreference chains, especially as anaphors. The only exception is the distribution of coordinated constructions in the validation data.[4] This

[4]Our validation data consist of just 3 texts. One of them is devoted to the "Kimmel and Short case". The same coordinated construction, "Kimmel and Short", is therefore used

Syntactic Structure	+anaphor		−anaphor		+ante		−ante	
+coordination	11	7%	141	93%	16	11%	136	89%
-coordination	1596	33%	3280	67%	1590	33%	3286	67%
+premodified	383	19%	1651	81%	520	26%	1514	74%
-premodified	1224	41%	1770	59%	1086	36%	1908	64%
+postmodified	277	22%	1010	78%	377	29%	910	71%
-postmodified	1330	36%	2411	64%	1229	33%	2512	67%
+postrestrictive	66	22%	231	78%	71	24%	226	76%
-postrestrictive	1541	33%	3190	67%	1535	32%	3196	68%

Table 4.6: Distribution of anaphors vs. non-anaphors (left part) and antecedents vs. non-antecedents (right part) for different syntactic structures in the training data (30 MUC-7 "dry-run" documents).

interaction reflects two factors: first, entities are normally introduced by more complex descriptions and then further referred to by simpler ones (Nenkova and McKeown (2003) have found the same tendency for proper names), and, second, parsing errors often result in NPs having very complex structure, but no linguistic relevance.

4.6 Intrasentential Constraints

Certain structural properties of a sentence may impose restrictions on possible coreference links within it. These restrictions have been one of major research topics within the transformational grammar framework, resulting in a number of formulations for the *non-coreference* rule, originally suggested by Langacker (1969). The rule stipulates contra-indexing, or non-coreference, of two NPs (NP_1,NP_2) if:

1. NP_1 is a pronoun. (This condition was proposed by Langacker, but then rejected in later studies, for example, (Lasnik, 1976; Reinhart, 1983).)

2. NP_2 is not a pronoun.

3. NP_2 is in the **domain** of NP_1.

Recall our Example (22B), repeated as (26) below. The second NP, "Zelda" is in the domain of the first NP, "She", and, therefore, coreference is prohibited:

(26) [She]$_1$ adores [Zelda]$_{2,ante\neq1}$'s teachers.

very often, leading to an unnatural distribution.

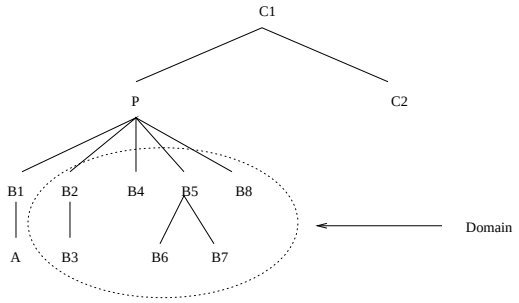


Figure 4.3: P-command relation, as defined by Barker and Pullum (1990).

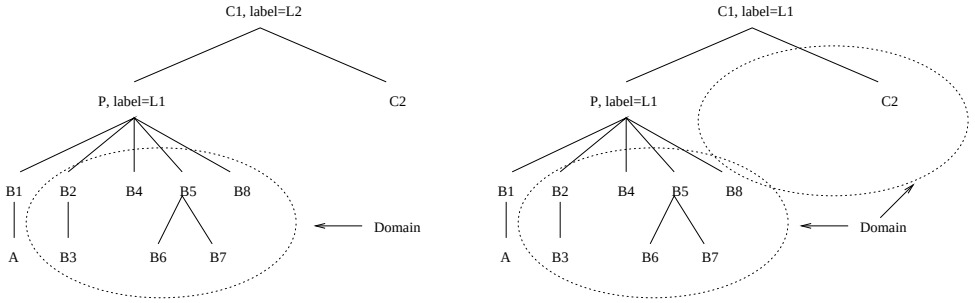


Figure 4.4: R-command relation, as defined by Reinhart (1983).

Various researchers proposed different definitions of domain in this context. Barker and Pullum (1990) show that these definitions can be described within a uniform theoretical framework of "command relations", defining domain as a set of nodes commanded by an NP in the parse tree and preceded by it in the surface string of the sentence.

We following Barker and Pullum (1990) and say that, for a property P, a node A in a parse tree P-commands a node B if and only if every node p that (1) properly dominates A and (2) satisfies condition P, also dominates B. Although this definition assumes that *every* node p should dominate B, it is sufficient to check that the *closest* node to A with property P dominates B (Latecki, 1991). Figure 4.3 shows an example: node A P-commands all the nodes A,P,$B_1..B_8$ and does not P-command nodes C_1 and C_2.

This definition generalizes over a number of command relations proposed in generative grammar studies:

C-command (generic definition): A node A C-commands a node B if the first branching node dominating A also dominates B.

S-Command (Langacker's "command"): A node A S-commands a node B if the first S-node dominating A also dominates B. S-node is a node in a parse tree tagged as a clause ("S", "SINV", "S1", or "SQ").

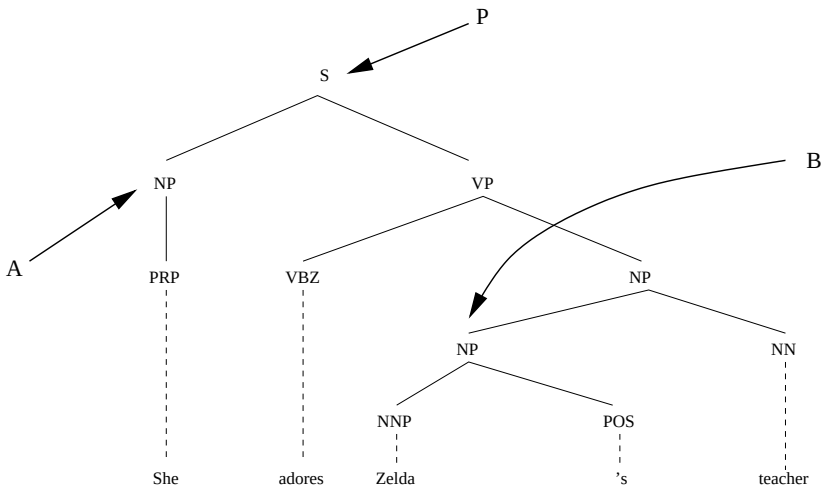


Figure 4.5: Contra-indexing stipulated by command relations.

R-Command (Reinhart's version of "ccommand"): A node A R-commands a node B if the branching node α_1 most immediately dominating A either dominates B or is immediately dominated by a node α_2 which dominates B, and α_2 is of the same category type as α_1. This is not a command relation in the sense of (Barker and Pullum, 1990), but still a very similar relation. Figure 4.4 shows two examples of R-command: on the left, the nodes α_1 (P) and α_2 (C_1) have different category types, and, thus, A only R-commands the nodes A, P, and $B_1..B_8$. On the right, the nodes α_1 (P) and α_2 (C_1) have the same category type (L_1), and, therefore, A R-commands all the nodes.

Figure 4.5 shows how the proposed definitions stipulate non-coreference in Example (22B): the markable "She" has the noun "Zelda" in its domain, and, thus, cannot corefer with it. Note that this case is covered by all the investigated command constraints. Reinhart (1983) shows a number of examples, where C-, S- and R-commands yield different analyses.

We have implemented C-command, S-command, and R-command in order to investigate, how useful the command-based contra-indexing constraints can be for our corpus. . The commands are represented as boolean features (for example, `ccommand(`$\mathtt{NP}_1$`,`$\mathtt{NP}_2$`)`), and computed straightforwardly from the parser's output.

Table 4.7 shows the distribution of the +coreferent vs. -coreferent pairs[5] satisfying and violating command constraints. The distribution is a bit surprising: our data contain numerous cases of coreference violating the contraindexing principle. In fact, R-command shows no interaction with coreference, and

[5]We only consider intrasentential anaphor in this section, so we only take into account pairs of markables within the same sentence.

command	All pairs				Non-pronominal anaphors			
	+coref		−coref		+coref		−coref	
−C-command	391	3%	11699	97%	182	2%	10846	98%
+C-command	223	4%	5264	96%	122	2%	4899	98%
−R-command	413	4%	11370	96%	205	2%	10555	98%
+R-command	201	3%	5593	97%	99	2%	5190	98%
−S-command	311	3%	9878	97%	148	2%	9119	98%
+S-command	303	4%	7085	96%	156	2%	6626	98%

Table 4.7: Distribution of coreference links for different command relations in the training data (30 MUC-7 "dry-run" documents).

command	All pairs				Non-pronominal anaphors			
	+coref		−coref		+coref		−coref	
−C-command	570	4%	14881	96%	281	2%	13786	98%
+C-command	44	2%	2082	98%	23	1%	1959	99%
−R-command	576	4%	15128	96%	282	2%	14021	98%
+R-command	38	2%	1835	98%	22	1%	1724	99%
−S-command	510	4%	12810	96%	250	2%	11834	98%
+S-command	104	2%	4153	98%	54	1%	3911	99%

Table 4.8: Distribution of coreference links for different modified command relations in the training data (30 MUC-7 "dry-run" documents)

S-command and C-command exhibit the opposite effect: violating Langacker's rule, markables are significantly (χ^2-test, $p < 0.01$) more often coreferent with the NPs within their domain than with those outside it. We have identified two main sources of command violation: apposition/copula and complex sentences.

According to the MUC definition of coreference, parts of apposition or copula construction should be coindexed, contrary to the assumptions of the transformational approach:

(27) [The crash]$_1$ Thursday was [the third Tomcat crash]$_{1,ante=2}$ this year.

Although the command theory accounts for at least some complex sentences (" Those who know [her]$_1$ adore [Zelda]$_2$.", example (22.C)), they are generally constructed by linguists and not representative of real data. Our corpus contains a lot of very long sentences with complex structure:

(28) She repeated [China]$_1$'s stance that [the tests]$_2$ were "routine and normal" while her deputy , Vice-Minister Shi Guangsheng , said [the test firings]$_{3,ante=2}$ were being conducted within [China]$_{4,ante=1}$'s territorial water.

Such sentences are problematic for the parser. Thus, in (28) the subordinate clause "while her deputy..." is erroneously attached to "the tests were...", putting NP_3 into the domain of NP_2 and NP_4 into the domain of NP_1.

This analysis raises several issues concerning the applicability of theoretical claims on coreference to the real-world data. First, theoretical research and data-oriented annotating guidelines may have different definitions of the problem. For example, parts of appositions, annotated in the MUC corpus, are not considered coreferent (and thus are not subject to any further analysis) by most theories. Second, theoretical claims are usually based on clean and correctly analyzed data. The accuracy may go down when they are applied to a corpus containing errors or when preprocessing with off-the-shelf NLP modules is required to obtain the relevant information. For example, parsing mistakes may affect the applicability of specific syntax-based predictions. Finally, theoretical predictions are made by inspecting some finite (and often small) amount of relevant data and may therefore have low coverage. For example, the command theory covers only the most simple patterns for complex sentences. Corpus-based analysis may help to adjust theoretical claims to a particular dataset. More systematic investigation of this problem is an important issue for future research, discussed in Section 9.2.

We have developed modified versions of the command functions to account for such cases. We say that a node A P-commands (modified) a node B if:

1. A P-commands B in the original sense (Barker and Pullum, 1990).

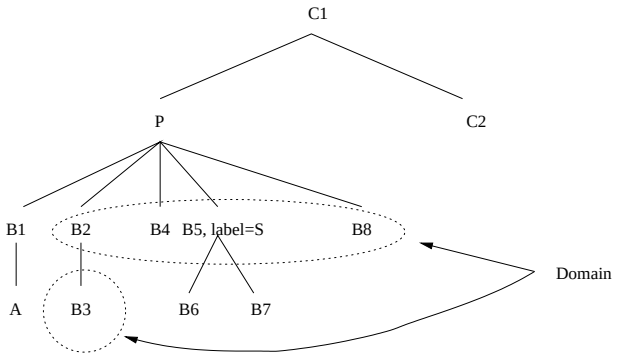


Figure 4.6: Modified P-command relation

2. A and B are not parts of an apposition or a copula construction (See section 4.7 for a detailed description).

3. The nodes p and B are in the same clause. From a computational perspective this means, that the path connecting nodes B and p in the parse tree does not contain any S-like ("S", "S1", "SINV", "SQ") nodes except for p itself.

Figure 4.6 shows an example of the modified P-command: the node A P-commands (modified) the nodes B_1, P, B_2, B_3, B_4, B_5, and B_8, but not the nodes B_6 and B_7, because there is an S-labeled node (B_5) between them and P.

Table 4.8 shows the distribution of ±coreferent pairs for the modified versions of commands. It suggests a clear interaction (χ^2-test, $p < 0.01$) between the command relations in their modified form and coreference. Nevertheless, we still see cases of coreferring markables violating the contraindexing principle.

In sum, the original definitions of command relations are not accurate enough for successful coreference resolution. These definitions, however, can be modified. Our new, more robust versions of commands, interact with coreference distribution and, thus, might provide useful information.

4.7 Explicit Indicators for Coreference

The above-mentioned syntactic constraints provide explicit evidence *against* coreference: if a command constraint is violated, two markables should not corefer. In this section we describe a complementary class of constraints — explicit syntactic indicators *for* coreference.

We have identified two structures of this kind: copula and appositive constructions:

(29) [About half the aircraft]$_1$ are [military planes]$_{2,ante=1}$ assigned to Air Force bases in Texas, Oklahoma, Kansas and New Mexico.

(30) [Plutonium]$_1$, [a highly radioactive element]$_{2,ante=1}$, causes cancer if inhaled.

We have already mentioned in Section 2.2, that linguistic theory would not describe such cases as "co-reference", because there is no true reference here. From an information extraction perspective, however, they are usually considered cases of coreference.

From a practical perspective, it is better to include such cases into a Coreference Resolution engine, as they help assemble together longer coreference chains. Consider the following snippet:

(31) ...the third [Tomcat]$_1$ crash this year...
[The Tomcat]$_{2,ante=1}$, [a $38 million, twin-engine aircraft]$_{3,ante=2}$ built by the Grumman Aerospace Corp., is [the standard fighter plane]$_{4,ante=3}$ used by the Navy for close-in air combat. [The aircraft]$_{5,ante=4}$ has been in operation since 1973. The oldest version of [the aircraft]$_{6,ante=5}$, the F-14A, is scheduled to be phased out by the year 2004.

The annotated chain contains two problematic markables: a second part of an apposition (M_3) and a predicative nominal (M_4). We see the same reasons for including them into the chain, discussed below on the example of M_3. Suppose a system can resolve (relatively easy) links $\{M_1, M_2\}$, $\{M_3, M_5\}$, $\{M_3, M_6\}$ and $\{M_5, M_6\}$. If we consider appositions to be cases of coreference, we can simply link M_3 to M_2, thus forming the chain $\{M_1, M_2, M_3, M_5, M_6\}$. If we do not consider appositions, we need much more intensive inference: first, we have to determine, that M_3 is a second part of an apposition and discard the $\{M_3, M_5\}$ and $\{M_3, M_6\}$ links. Second, in order to produce the desired chain ($\{M_1, M_2, M_5, M_6\}$, the same as before, but without M_3), we have to link M_5 to either M_1 or M_2, which is obviously a more difficult task than linking M_5 to M_3: we now need sophisticated domain knowledge to match "aircraft" to "Tomcat".

In sum, if we do not accept appositives and copulas as cases of coreference, we still have to identify these construction (to discard non-referring NPs, such as M_3 in our example), and, in addition, we need much more complex inference to resolve subsequent markables.

Several researchers (Baldwin et al. 1997; Soon et al. 2001, among others) have incorporated apposition and copula into their systems. However, most of them reported unexpectedly moderate performance, because these constructions could not be identified with high precision. Thus, the system of Soon et al. (2001) achieves 2.4% recall and 60.0% precision on the MUC-7 test data,

using just the "appositive" feature. For the same data, Baldwin et al. (1997) report 3.3% recall and 64.0% precision for the combination of apposition and copula. Both studies claim that they have anticipated a much higher precision lever.

We have tried to develop more sophisticated heuristics for identifying appositions and copulas to avoid the same precision loss.

Apposition. First, we extract candidates for appositions using a regular expression matcher ($[[.+]_{NP1}$, $[.+]_{NP2}$,?$]_{NP}$). As most approaches only use this information, we have encoded it as the `apposition_basic` feature.

Data analysis (see the upper part of Table 4.9) shows that more than half of candidate appositions are false positives: parts of these structures are not annotated as coreferent. We identify and discard patterns, syntactically similar to appositions, but not indicating for coreference. First, we check if our candidate is a part of `coordination` (see Section 4.5). Second, we identify two other types of similar-looking non-appositive constructions observed in the training data:

(32) address: The Federal Aviation Administration underestimated the number of aircraft flying over the Pantex Weapons Plant outside $[\text{Amarillo}]_1$, $[\text{Texas}]_{2,ante\neq1}$, where much of the nation's surplus plutonium is stored, according to computerized studies under way by the Energy Department.

(33) age/time: $[\text{Washington}]_1$, $[\text{Feb. } 22]_{2,ante\neq1}$ (Bloomberg) – The Navy ordered its Northrop Grumman Corp. F-14s out of the skies for three days while it investigates three recent F-14 crashes.
$[\text{Reid}]_1$, $[52]_{2,ante\neq1}$, who for nearly 27 years was a broker at the Dean Witter Reynolds office in Palo Alto, was president of the Half Moon Bay Pilots Association.

For *addresses*, we check if the head noun of NP_1 and NP_2 are proper names. If at least one of them is marked as LOCATION by the NE-tagger, we discard the candidate. For *time/age*, we check if NP_2's head is a number (string of digits).

We filter the set of candidate appositions, discarding appositive-coordinate constructions, addresses and age/time descriptions. The resulting subset is encoded with the `apposition` feature.

Table 4.9 shows the distribution of coreference links for the basic and modified apposition features. Both of them interact with coreference (χ^2-test, $p < 0.01$), but the modified version results in a much better prediction accuracy (3-4 times less {+apposition, −coreferent} cases). The errors stem from the parsing mistakes, especially when analyzing structures combining appo-

Construction	+coreferent		−coreferent	
+apposition_basic	63	42%	88	58%
−apposition_basic	551	3%	16875	97%
+apposition	60	73%	22	27%
−apposition	554	3%	16941	97%

Table 4.9: Distribution of coreference links for different apposition features in the training data (30 MUC-7 "dry-run" data), only intrasentential coreference is considered.

sition and coordination, as in the example (23) above: inconsistent parsing makes it problematic to identify and discard all such cases.

The MUC guidelines say that two parts of an apposition should always be considered coreferent (provided the second NP is not negated, see Section 2.2). The data show that this syntactic construction can be identified reliably even with an error-prone parsing module and that its distribution interacts with coreference. A resolution engine could therefore benefit from accounting for appositions. We will see in Section 8.2 how the inaccuracies of our `apposition` feature affect the overall system performance.

Predicate Nominals. According to the MUC annotation scheme (Hirschman and Chinchor, 1997), predicate nominals are typically coreferent with the copula subject:

(34) [Bill Clinton]$_1$ is [the President of the United States]$_{2,ante=1}$.

According to the MUC guidelines, coreference should not be recorded if the text only asserts the possibility of identity between two markables (35), in the case of negation (36) or a partial set overlap[6] (37):

(35) If elected, [Phinneas Flounder]$_1$ would be [the first Californian in the Oval Office]$_{2,ante\neq1}$.

(36) [Mediation]$_1$ is not [a viable alternative to bankruptcy]$_{2,ante\neq1}$.

(37) [Mediation]$_1$ is often [a viable alternative to bankruptcy]$_{2,ante\neq1}$.

Unfortunately, the picture gets more complicated when time-dependent coreference links are concerned. Generally, two markables should be recorded

[6]The same restrictions apply to apposition, but they are not relevant for our approach, because the parser does not identify these constructions ("[The criminals]$_1$, [often legal immigrants]$_{2,ante\neq1}$") as candidate appositions.

as coreferential if the text asserts them to be coreferential at ANY TIME:

(38) [Henry Higgins]$_1$, who was formerly [sales director for Sudsy Soaps]$_{2,ante=1}$, became [president of Dreamy Detergents]$_{3,ante=1}$.

In some cases, however, annotators should cut undesired "outdated" links to prevent the collapsing of coreference chains:

(39) [Henry Higgins]$_1$, who was formerly [sales director for Sudsy Soaps]$_{2,ante=1}$, became [president of Dreamy Detergents]$_{3,ante=1}$. Sudsy Soaps named [Eliza Dolittle]$_4$ as [sales director]$_{5,ante=4,\neq 2}$ effective last week.

(40) [The stock price]$_1$ fell from [$4.02]$_{2,ante\neq 1}$ to [$3.85]$_{3,ante=1}$.

These rules, as formulated in the MUC guidelines, are very vague and problematic even for human annotators (Hirschman et al., 1997), causing a lower-than-expected inter-annotator agreement for the MUC corpora (see also the discussion in Section 2.2). Hirschman and Chinchor (1997) mention that this part of the guidelines should be revisited in the future.

We have investigated several functions to automatically identify copulas. First, we use a regular expression matcher to find and analyze predicative constructions: we check if the antecedent is directly dominated by a node α and the path from α to the anaphor only contains VP-labeled nodes. We then divide the VP material preceding the antecedent into the main verb (the closest word labeled as AUX, VBD, VBN, VB, VBP, VBZ or VBG) and the auxiliary part (possibly containing modal and auxiliary verbs, adverbs and negation). The following main verbs indicate copula constructions: "be", "become", and "call" (only in the form "called" labelled VBN). An example is shown on Figure 4.7.

To avoid "outdated" links, we first only identify predicative constructions in the present (with "am", "are", "is", "'s", "become", "becomes", and "called" as the main verbs). This information is encoded in the `copula_present` feature: it is set to 1 if an anaphor and its antecedent are parts of a predicative construction in the present tense and to 0 otherwise.

The `copula_present` feature seems to be too restrictive: it only describes a dozen of coreference links in the whole corpus. Therefore we have developed another function, `copula_all`, allowing for all morphological forms of the main verb. A more refined version, `copula_all_notmodal` combines the `copula_all` function with a simple check for modal and negative constructions: if the auxiliary part of a predicate contains "could", "would", "might", or a negation (an RB-labelled word), it is discarded.

Table 4.10 shows the distribution of coreferring vs. non-coreferring links for different variants of copulas. As we see, the `copula_present` feature is a

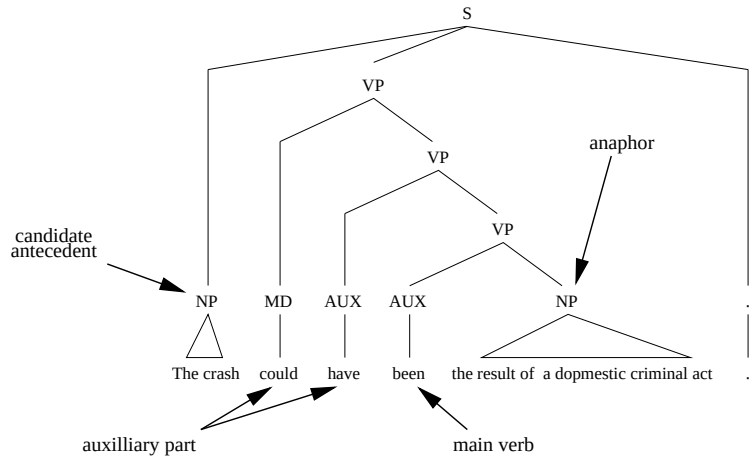


Figure 4.7: An example of a predicative construction.

very poor predictor of coreference: it almost never fires and the prediction's accuracy is low (less than 50% of +copula_present links are coreferential). The `copula_all` function fires for three times more links, but the accuracy is even lower. Although the precision level increases for the refined version, `copula_all_notmodal`, the error is still very high.

Analyzing the training data, we have found the following problematic cases, where parts of copula constructions are not coreferent:

(41) "There" constructions: $[\text{There}]_1$ is $[\text{little doubt}]_{2,ante\neq 1}$ about the admiral's combat credentials.

(42) Place/time modifiers: $[\text{It}]_1$ has been $[\text{14 years}]_{2,ante\neq 1}$ since a hijacked airplane has landed in Britain.

(43) Expletive "it": We have said $[\text{it}]_1$ is $[\text{our strategy}]_{2,ante\neq 1}$ to grow Provident .

We use surface matching to identify "there" constructions. For place/time modifiers, we check if the anaphor is a proper name tagged as TIME, DATE, or LOCATION. Finally, we have implemented simple rules to detect expletive pronouns in the experiments reported in this chapter. The task of determining non-anaphoric markables is addressed more systematically in Chapter 7 below.

We combine the `copula_all_notmodal` feature with these filters to obtain the final `copula` value. Table 4.10 suggests, that, although all the variants of copulas interact with the coreference distribution (χ^2-test, $p < 0.01$), the final `copula` is the best predictor. The remaining errors comprise mainly parsing mistakes, some cases of "partial overlap" mentioned in the annotation

Construction	+coreferent		−coreferent	
+copula_present	13	46%	15	54%
−copula_present	596	4%	16425	96%
+copula_all	39	43%	52	57%
−copula_all	570	3%	16388	97%
+copula_all_notmodal	35	45%	42	55%
−copula_all_notmodal	574	3%	16398	97%
+copula	35	60%	23	40%
−copula	574	3%	16417	97%

Table 4.10: Distribution of coreference links for different copula features in the training data (30 MUC-7 "dry-run" documents), only intersentential coreference considered

guidelines (see example (37)), and constructions with adjective-formed NPs ("[The results] are just [the opposite]").

To summarize, appositions and predicate nominals are important indicators of coreference. Unfortunately, identifying these constructions automatically is a very non-trivial task and the prediction quality is therefore only moderate.

4.8 Grammatical Roles

Most pronoun resolution algorithms incorporate grammatical role information. On the one hand, grammatical roles are needed to compute various salience parameters (see Section 6 below). On the other hand, several researchers, for example Mitkov (1998), Kennedy and Boguraev (1996), and Preiss (2001) have argued for the importance of syntactic parallelism for anaphora interpretation. In our system we have implemented several features to encode or rely on grammatical roles.

Most anaphora resolution algorithms (for example, Strube et al. (2002a)) assume that grammatical roles of NPs are given. Unfortunately, there is no way of computing them straightforwardly from a shallow parser's output — a bracketing structure augmented with very simple constituent labels (S, NP, NNP,...). Blaheta and Charniak (2000) suggest a statistical algorithm for extracting grammatical roles from shallow parsed data.

In our experiments we adopt a simplified model: the grammatical role of an NP depends only on the part-of-speech tag of its predecessor constituent. In most cases the predecessor is the node directly dominating the NP. An example is shown in Figure 4.8, with dotted arrows pointing from markable nodes to their predecessors. In the example sentence we have five markables: "We" is a *subject* (descendant of S), "people's lives" is an *object* (descendant of VP), "our hands" is a *complement* of a preposition, and "people" and "our"

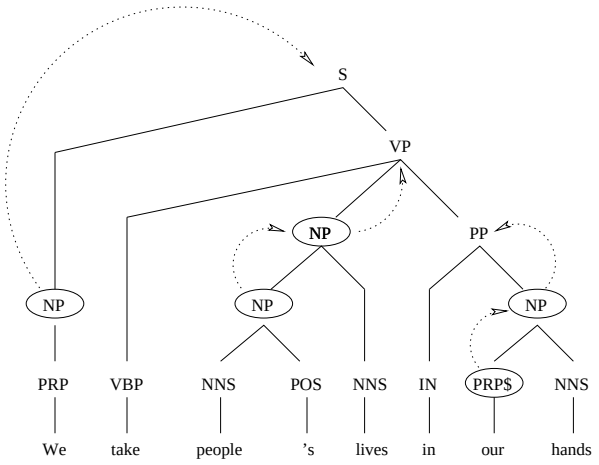


Figure 4.8: Assigning grammatical roles to the parser's output.

are *modifiers* (descendants of NP).

We go one level further up the tree if the node is a part of an appositive or coordinate structure or a main part of a complex NPs:

(44) A. [[A student pilot]$_{NP}$, [Rick DeLisle]$_{NP}$]$_{NP}$, told CNN of witnessing the crash.
B. [[A student pilot]$_{NP}$ and [his instructor]$_{NP}$]$_{NP}$ told CNN of witnessing the crash.
C. [[A student pilot]$_{NP}$ flying a small plane in the area]$_{NP}$ told CNN of witnessing the crash.

In all these cases the first NP, "a student pilot", is a subject (descendant of S), and not a modifier, although it is directly dominated by an NP-node.

Grammatical roles can be computed with different degrees of granularity. The classification adopted in this thesis is shown in Table 4.11. The roles are encoded as a nominal feature (`grammatical_role`) or as four binary ones (`subject`, `object`, `pp_complement`, `modifier`). For some markables (around 1.4%) we were not able to automatically assign any grammatical role: their predecessor nodes were not labelled as VP, PP, NP, or S-like.

We have manually annotated the validation data with this information to assess the accuracy of our automatically extracted grammatical roles, The right columns of Table 4.11 show that each role can be assigned automatically with around 90% precision and recall. The remaining 10% arise mainly from parsing errors: either the markable itself results from a bracketing error, or it is linked to a wrong position in the tree.

Grammatical roles are primarily used for determining accessibility and salience ranking of candidate antecedents (see Section 6, in particular, 6.5).

Role	Predecessor's tag	Performance		
		Recall	Precision	F-measure
Subject	S, S1, SINV, SQ	92.7	95.2	94.0
Object	VP	94.5	89.7	92.0
PP complement	PP	97.6	89.9	93.6
Modifier	NP	88.9	90.4	89.6

Table 4.11: Extracting grammatical roles from the parser's output: performance on the validation set (3 MUC-7 "formal train" documents).

The highest ranked role is a subject, and it is given a high preference when searching for an antecedent. This means that it is important for a salience-based approach to identify subjects accurately.

We have therefore developed two additional features for subject NPs. The `sentence_subject` is identified as a subject node dominated only by NP and S-like nodes. Sentence subjects are supposed to be the most salient entities. We also compute `minimal_depth_subject` — the markable with the smallest `depth`. Depth is measured as the distance from the root in the parse tree. For a named entity, we measure the depth of its first word.

Table 4.12 shows the distribution of anaphors vs. non-anaphors and antecedents vs. non-antecedents for different grammatical roles. The χ^2-test suggests a strong interaction ($p < 0.01$) between coreference properties of a markable, in particular, the probability of it being an antecedent, and its grammatical role. The interaction is exhibited for both the full set of grammatical roles and the special features for determining the subject, with the only exception for the distribution of anaphors vs. non-anaphors for ±`minimal_depth_subject` on the training data. This confirms the assumptions of salience-based research on anaphora that grammatical roles impose restrictions on the accessibility of the markables and, thus, influence their coreference properties. We will come back to grammatical roles in Section 6.6.

Syntactic parallelism. Comparing grammatical functions of an anaphor and an antecedent, we can account for syntactic parallelism. Consider the following example from Mitkov (1999):

(45) A. The programmer successfully combined $[\text{Prolog}]_1$ with $[\text{C}]_2$, but he had combined $[\text{it}]_{3,ante=1}$] with Pascal last time.
B. The programmer successfully combined $[\text{Prolog}]_1$ with $[\text{C}]_2$, but he had combined Pascal with $[\text{it}]_{3,ante=2}$] last time.

In this case, syntactic parallelism helps choose the correct antecedent for the markable "it". Unfortunately, it is only a very weak indicator that can be

Role	Anaphors				Antecedents			
	+anaphor		−anaphor		+ante		−ante	
SUBJECT	174	52%	162	48%	180	54%	156	46%
OBJECT	44	25%	132	75%	40	23%	136	77%
PP COMPLEMENT	74	22%	256	78%	77	23%	253	77%
MODIFIER	68	57%	52	43%	63	53%	57	47%
NONE	3	21%	11	79%	3	21%	11	79%
+subject	174	52%	162	48%	180	54%	156	46%
-subject	189	30%	451	70%	183	29%	457	71%
+sentence_subject	127	53%	112	47%	134	56%	105	44%
-sentence_subject	236	32%	501	68%	229	31%	508	69%
+minimal_depth_subject	89	51%	87	49%	99	56%	77	44%
-minimal_depth_subject	274	34%	526	66%	264	33%	536	67%

Table 4.12: Distribution of anaphors vs. non-anaphors (left part) and antecedents vs. non-antecedents (right part) for different grammatical roles in the training data (30 MUC-7 "dry-run" documents).

overridden by a number of other constraints or preferences, as another example from Mitkov (1999) shows:

(46) A. Vincent removed [the diskette]$_1$ from [the computer]$_2$ and then copied [it]$_{3,ante=1}$.
B. Vincent removed [the diskette]$_1$ from [the computer]$_2$ and then disconnected [it]$_{3,ante=2}$.

In both sentences here, the pronoun "it" has the same grammatical role as "the diskette". However, in sentence (B) such anaphoric binding is not licensed by the semantic consistency constraints.

The importance and frequency of parallel constructions depends crucially on a text's genre. According to Mitkov (1998), parallelism is a good predictor of coreference for technical manuals. Lapin and Leass (1994), on the contrary, report virtually no performance loss on their dataset after discarding parallelism-related information. We have found only very few parallel constructions in our corpus, a collection of newswire texts, and, therefore, we provide only very simple analysis of this phenomenon.

We say that two markables are `parallel`, if they have the same grammatical roles and appear in the same or adjacent sentences. This information is obviously much more important for pronominal anaphors than for any other kind of markables, therefore we have implemented the `parallel_pronoun_ana`

Parallelism	Links			
	+coreferent		−coreferent	
+parallel, all the pairs	254	5%	5081	95%
−parallel, all the pairs	1149	3%	40847	97%
+parallel, anaphor is a pronoun	117	30%	267	70%
−parallel, anaphor is a pronoun	492	12%	3549	88%

Table 4.13: Distribution of coreference links for syntactic parallelism features in the training data (30 MUC-7 "dry-run" documents), only pairs from the same or adjacent sentences are considered.

feature: it is set to 1 if the anaphor and the antecedent are `parallel` and the anaphor is a pronoun.

Table 4.13 shows the distribution of coreference links satisfying and violating the parallelism preference for all pairs (upper part) and for cases where the anaphor is a pronoun (lower part). The χ^2-test suggests an interaction ($p < 0.01$) between parallelism and coreference. We see, at the same time, many cases of non-coreferential parallel markables, making the `parallel` and `parallel_pronoun_ana` features rather weak indicators for coreference on our data.

To summarize, in this Section we have presented our features encoding grammatical roles and syntactic parallelism. The χ^2-test shows that they interact with the coreference properties of markables. These features, however, do not impose any hard conditions on the anaphoricity, but indicate important preferences. Thus, subjects more often tend to be antecedents than other NPs and parallel markables more often tend to be coreferential than non-parallel. In Chapter 6 we present and discuss several salience-related features based on grammatical roles.

4.9 Morphological Agreement

All markables in a coreference chain refer, by definition, to the same entity and, consequently, should share same number and person characteristics. Agreement information is often used as a filter: if two markables disagree in, for example, number, a typical coreference resolution system suggests that they belong to different chains. The MUC-7 data, however, contain examples, sometimes not very intuitive, of coreferring markables with different number and/or person values:

(47) He even refused to let [the crew]$_1$ watch television , fearing it might distract [them]$_{2,ante=1}$ and make [their]$_{3,ante=1}$ jobs even harder.

(48) Tom Coonan had been at sea less than two days when his U.S. Coast Guard cutter was forced by [rough weather]$_1$ to abandon its search Friday for the remains of TWA Flight 800... Search and rescue efforts were hampered Friday by [rain and heavy seas]$_{2,ante=1}$.

(49) [American Airlines]$_1$, for example, has told families of the people who died when one of [its]$_{2,ante=1}$ planes crashed in Colombia in December that [it]$_{3,ante=1}$ would negotiate as if the limit did not exist.

(50) "This doesn't surprise [me]$_1$ at all," said [Trish Neusch]$_{2,ante=1}$...

(51) [Provident]$_1$ Vice President Thomas White said , "[We]$_{2,ante=1}$ have said it is [our]$_{3,ante=1}$ strategy to grow [Provident]$_{4,ante=1}$".

In the first sentence, we see a very common case of number disagreement: though "the crew" is syntactically singular, it denotes a group entity and, thus, can be further referred to as "them". As example (48) shows, defining "group entity" is a very non-trivial task, as almost every NP can be interpreted as a group in an appropriate context. In the sentence (49), we see a plural expression "American Airlines", but the NP as a whole is a name, and, thus, should be analyzed as singular. Our example (50) shows a typical disagreement in person: pronouns receive different interpretation for different speakers (Trish Neusch and the document's author in our example). Finally, all these factors can interplay, as in the last example: "we" and "Provident" have both different number and person.

Theoretical studies (Kamp and Reyle 1993, among others) raise a number of issues with plural anaphora. This research area involves providing accurate treatment for different kinds of plural constructions. It often implies complex linguistic modeling. Kamp and Reyle (1993) identify several "processes of antecedent formation" for personal plural pronouns: for example, summation ("John took Mary to Acapulco. They had a lousy time.") or abstraction ("Susan has found every book which Bill needs. They are on his desk"). Deep analysis of plural descriptions is a challenging problem, lying outside the scope of this thesis.

Rule-based systems for anaphora resolution (from the naive syntactic algorithm of Hobbs (1978) to more recent centering based systems, discussed in Chapter 6) typically rely on hard agreement constraints, and, thus, are not able to successfully analyze problematic disagreement cases. We encode agreement information as features, allowing for more flexibility.

We compute the agreement characteristics of a markable by examining its head. If the parser has assigned a noun tag to the head, we classify the markable as 3rd person singular (NN, NNP, CD), third person plural (NNS, NNPS), or third person unknown number (any other tag). If the head is

Agreement feature	Links			
	+coreferent		−coreferent	
+same_number	7064	3%	272462	97%
−same_number	371	0%	156179	100%
+same_person	6974	2%	412750	98%
−same_person	461	3%	15891	97%
+same_person_quoted	7373	2%	425768	98%
−same_person_quoted	62	2%	2873	98%

Table 4.14: Distribution of coreference links for different syntactic agreement features in the training data (30 MUC-7 "dry-run" documents).

a pronoun, we consult our list of preclassified pronouns. Coordinations are always third person plural. If two markables share a syntactic property (but not the "unknown" value"), we set the corresponding agreement feature to 1. The ambiguous pronouns "you" and "your" agree in number with both singular and plural markables.

Apart from the standard agreement features, we have implemented a relaxed version of person agreement, `same_person_quoted`, saying that two markables agree if they have the same person or at least one of them is a part of a quoted string: most cases of disagreement in person result from the speaker change (see examples (50) and (51)). Conjoining the `same_number` and `same_person` (or `same_person_quoted`) features, we obtain `syntactic_agreement` and `syntactic_agreement_quoted` values.

Table 4.14 shows the distribution of coreference links for agreeing and disagreeing pairs. The χ^2-test suggests that both the `same_number` and `same_person` features significantly affect coreference distribution ($p < 0.01$). For the `person_quoted` value we get no effect on the training data.

At the chains level, 11.3% (18.9%) of the chains in the training (validation) data contain at least one pair of markables with different number and 5.6% (13.3%) — at least one pair with different person. Only 84.8% (71.1%) of chains do not violate any syntactic agreement constraints. If we allow disagreement in person in quoted sentences, 0.8% (2.2%) of chains still have inconsistent `person` values and 11.7% (21.1%) show at least some syntactic disagreement.

In Section 5 below we discuss gender and semantic class agreement.

4.10 Experiments

In this Chapter we have seen that various syntactic parameters significantly affect distributions of ±anaphors, ±antecedents, and ±coreferential links. This

suggests that syntactic evidence might be beneficial for coreference resolution. In the present Section we encode syntactic knowledge as 61 features to build a learning-based coreference resolution system[7].

Our syntactic features are listed in Table 4.15. As an illustration, we also show their values for the pair ("a viable alternative", "Mediation") in the following example:

(52) [Mediation]$_1$ is not [a viable alternative to bankruptcy]$_{2,ante\neq1}$.

We have seen throughout this Chapter, that not all the features are equally useful. Thus, head-related features may potentially increase overfitting, and for commands, appositions and copulas we have basic and more sophisticated heuristics. These less relevant "secondary" features are shown in italic.

Most syntactic constraints operate on intrasentential level. Therefore same-sentence coreference (Experiment 3) is an ideal testbed for a purely syntactic approach. In Experiment 4 we rely on syntactic evidence for full-scale NP-coreference resolution.

4.10.1 Experiment 3: Syntactic Knowledge for Intrasentential Anaphora Resolution

In this Experiment, our validation data, 3 MUC-7 "formal train" documents, have been used for testing. We removed from both the training and validation sets anaphoric links spanning over sentence boundaries.[8]

Baseline. As everywhere throughout the thesis, we have used the system of Soon et al. (2001) as a baseline. It has achieved an F-score of 39.7% (29.6% recall and 62.5% precision) on the same-sentence coreference task. This result is surprisingly low: on average, the baseline achieves 54.5% recall on the validation data. There are two possible reasons for the low recall here: first, we train our classifier only on intrasentential pairs. This drastically decreases the amount of training material. Second, the most informative Soon et al.'s (2001) features, `match` and `alias`, are surface-oriented. Consequently, the baseline performs very well on similar expressions. Within the same sentence, however, different descriptions are normally used:

[7] To run the SVM^{light} learner, we have converted nominal features into boolean ones. This resulted in 109 features.

[8] A system, that correctly resolved all the remaining (i.e., intrasentential) anaphors, would achieve 26.3% recall in the full-scale evaluation on the validation data.

Table 4.15: Syntactic features and their values for the pair ("a viable alternative", "Mediation") in Example (52). Secondary features are shown in italic.

Feature	Range	Example value
Anaphor's parameters		
type_of_markable(M_i)	DEF,NE,PRON,OTHER	OTHER
type_of_pronoun(M_i)	PERS, POSS, REFL, NONE	NONE
type_of_definite(M_i)	THE, DT, NONE	NONE
determiner(M_i)	nominal	a
det_ana_type(M_i)	ANA, NONANA, OTHER	NONANA
det_ante_type(M_i)	ANTE, NONANTE, OTHER	OTHER
head_anaphoric(M_i)	0,1	0
head_nonanaphoric(M_i)	0,1	0
head_antecedent(M_i)	0,1	0
head_nonantecedent(M_i)	0,1	0
coordination(M_i)	0,1	0
premodified(M_i)	0,1	1
postmodified(M_i)	0,1	1
postrestrictive(M_i)	0,1	0
grammatical_role(M_i)	SUBJ, OBJ, MOD, PP_COMPL, MOD, NONE	OBJ
subject(M_i)	0,1	0
sentence_subject(M_i)	0,1	0
minimal_depth_subject(M_i)	0,1	0
number(M_i)	SG, PL, AMB, UNKNOWN	SG
person(M_i)	1,2,3	3
Antecedent's parameters		
type_of_markable(M_j)	DEF,NE,PRON,OTHER	OTHER
type_of_pronoun(M_j)	PERS, POSS, REFL, NONE	NONE
type_of_definite(M_j)	THE, DT, NONE	NONE
determiner(M_j)	nominal	0
det_ana_type(M_j)	ANA, NONANA, OTHER	NONANA
det_ante_type(M_j)	ANTE, NONANTE, OTHER	NONANTE
head_anaphoric(M_j)	0,1	0
head_nonanaphoric(M_j)	0,1	0
head_antecedent(M_j)	0,1	0
head_nonantecedent(M_j)	0,1	0
coordination(M_j)	0,1	0
premodified(M_j)	0,1	0
postmodified(M_j)	0,1	0
postrestrictive(M_j)	0,1	0

Table 4.15: (continued)

Feature	Range	Example value
grammatical_role(M_j)	SUBJ, OBJ, MOD, PP_COMPL, MOD, NONE	SUBJ
subject(M_j)	0,1	1
sentence_subject(M_j)	0,1	1
minimal_depth_subject(M_j)	0,1	1
number(M_j)	SG, PL, AMB, UNKNOWN	SG
person(M_j)	1,2,3	3
Pair's parameters		
ccommand(M_i, M_j)	0,1	1
scommand(M_i, M_j)	0,1	1
rcommand(M_i, M_j)	0,1	1
ccommand_modified(M_i, M_j)	0,1	0
scommand_modified(M_i, M_j)	0,1	0
rcommand_modified(M_i, M_j)	0,1	0
apposition_basic(M_i, M_j)	0,1	0
apposition(M_i, M_j)	0,1	0
copula_present(M_i, M_j)	0,1	1
copula_all(M_i, M_j)	0,1	1
copula_all_notmodal(M_i, M_j)	0,1	0
copula(M_i, M_j)	0,1	0
same_number(M_i, M_j)	0,1	1
same_person(M_i, M_j)	0,1	1
same_person_quoted(M_i, M_j)	0,1	1
synt_agree(M_i, M_j)	0,1	1
synt_agree_quoted(M_i, M_j)	0,1	1
parallel(M_i, M_j)	0,1	0
parallel_pronoun(M_i, M_j)	0,1	0

Features	Validation set		
	Recall	Precision	F
Soon et al. (2001), SVM baseline	29.1	62.5	39.7
syntactic	††14.6	48.4	22.4
syntactic+Soon et al. (2001)	35.9	58.7	44.6

Table 4.16: A syntactic approach (pure and combined with Soon et al.'s (2001) features) to same-sentence coreference: performance on the validation (3 MUC-7 "train" documents) data. Significant improvements over the main baselines (SVM^{light} learner, features of Soon et al. (2001)) are shown by */** and significant losses — by †/†† ($p < 0.05/p < 0.01$).

(53) In the midst of last year's commemorations of the 50th anniversary of the end of World War II, the Kimmel family asked the Pentagon to restore [Kimmel and Short]$_1$ to [their]$_{2,ante=1}$ highest ranks posthumously as symbolic recognition that [they]$_{3,ante=2}$ had been made [scapegoats]$_{4,ante=3}$ for the mistakes of others.

In (53), the same entity is mentioned four times: "Kimmel and Short", "their", "they", and "scapegoats".

In all the tables in this section we show significant improvements over the main baselines (SVM^{light} learner, features of Soon et al. (2001)) */** and significant losses — by †/†† ($p < 0.05/p < 0.01$).

Syntactic features. Table 4.16 summarizes the performance figures of our syntactic features for intrasentential coreference resolution. Once again, we see very low recall figures: most syntactic features provide evidence against coreference, and the only positive indicators, appositions and copulas, represent very specific constructions and cannot account for the majority of anaphors. As far as the distinction between main and secondary features is concerned, we have not observed any significant difference (the F-scores reported in the Table correspond to "main" features, with "secondary" features omitted).

The last line of Table 4.16 shows the performance figures for the combination of syntactic knowledge with the 12 "basic" features of Soon et al. (2001). Compared to the baseline, we see a slight non-significant gain in recall (the approach of Soon et al. (2001) does not account for copula), leading to 8.1% relative improvement in F-score.

	Test set			Validation set		
Features	Recall	Prec.	F	Recall	Prec.	F
Baselines						
"merge all"	86.6	35.2	50.0	91.9	38.0	53.7
basic features	50.5	75.3	60.4	54.5	56.9	55.7
Syntactic features						
syntax	††7.6	68.5	13.8	††9.9	57.4	16.9
Other knowledge sources						
matching	52.2	††61.2	56.3	56.2	53.3	54.7

Table 4.17: A pure syntactic approach to the full-scale Coreference Resolution task: performance on the testing (20 MUC-7 "formal test" documents) and the validation (3 MUC-7 "train" documents) data. Significant improvements over the main baselines (SVM^{light} learner, features of Soon et al. (2001)) are shown by */** and significant losses — by †/†† ($p < 0.05/p < 0.01$).

4.10.2 Experiment 4: Syntactic Knowledge for Full-Scale Coreference Resolution

Pure syntactic approach. As Table 4.17 shows, a pure syntax-based approach achieves only a low recall level. We have already observed the same tendency in Experiment 3: with almost no positive indicators, a syntactic algorithm does not have enough evidence to resolve most anaphors.

The suggested links, however, are reliable: although the classifier's precision (68.5%) is lower than the baseline's, it is still higher than the corresponding figures for all other knowledge types (see also Experiments 6 and 8).

In sum, a pure syntax-based system can only resolve very few anaphors, but the suggested links are very accurate.

Combining Syntax with the Basic Coreference Features. Table 4.18 shows the performance figures for the combination of syntactic (Table 4.15) and "basic" (Soon et al., 2001) features. Augmenting the baseline with syntactic evidence, we have achieved a relative F-score improvement of 3.3%.

At first glance, it might look surprising: our syntactic features encode mostly contra-coreference constraints and, therefore, should help improve precision by prohibiting erroneous links. However, as we have noted in the previous Experiment, the baseline system usually suggest intersentential antecedents, relying mainly on matching. Syntactic constraints, on the contrary, operate mainly on the intrasentential level and, therefore, cannot help in this case.

Features	Test set Recall	Prec.	F	Validation set Recall	Prec.	F
Baselines						
"merge all"	86.6	35.2	50.0	91.9	38.0	53.7
basic features	50.5	75.3	60.4	54.5	56.9	55.7
Syntactic+basic features						
basic+syntax	52.2	75.2	61.7	56.2	56.5	56.4
Other knowledge sources						
basic+matching	**58.4	††63.1	60.6	60.6	54.1	57.1
basic+syntax	52.2	75.2	61.7	56.2	56.5	56.4

Table 4.18: Basic coreference resolution algorithm Soon et al. (2001) augmented with syntactic knowledge, performance on the testing (20 MUC-7 "formal test" documents) and the validation (3 MUC-7 "train" documents) data. Significant improvements over the main baselines (SVM^{light} learner, features of Soon et al. (2001)) are shown by */** and significant losses — by †/†† ($p < 0.05/p < 0.01$).

4.11 Summary

In this Chapter we have investigated the influence of syntactic knowledge on coreference resolution: types of markables (Section 4.2), determiners (Section 4.3), NP heads (Section 4.4), specific syntactic constructions (Section 4.5), command relations (Section 4.6), appositions and copulas (Section 4.7), grammatical roles and syntactic parallelism (Section 4.8) and syntactic agreement (Section 4.9).

We have presented several data-driven modifications of syntactic rules for coreference resolution, leading to more robust and accurate features. However, even our more robust versions of syntactic rules are not error-free, and, therefore, cannot be seen as hard constraints. We represent this knowledge as features in a machine learning-based coreference resolution algorithm, thus, treating it rather as preferences.

Altogether we have encoded 61 syntactic features (108 boolean) and run two SVM^{light}experiments: for intrasentential anaphora resolution (Experiment 3, Section 4.10.1) and full-scale coreference (Experiment 4, Section 4.10.2). Both experiments show that, on the one hand, syntactic knowledge is reliable, but, on the other hand, helps resolve only few anaphors. In combination with Soon et al.'s (2001) features, syntactic constraints show a slight improvement over the baseline.

In the following two Chapters, we will investigate deeper kinds of information — semantic and discourse evidence.

Chapter 5

Semantic Compatibility

In the previous two chapters we have investigated shallow matching and syntactic approaches to coreference resolution. We have seen that adding such knowledge to a baseline system (Soon et al., 2001) results in slight performance gains. However, there is still room for improvement, and we expect our system to benefit from deeper information — semantic (present Chapter) and discourse (Chapter 6) knowledge. Our position follows numerous linguistic studies (see Sections 5.1 and 6.1) claiming that semantic and discourse knowledge ought to help a coreference resolution algorithm.

Soon et al. (2001) point out that 63% of the recall errors made by their system were due to the lack of information and inadequacy of surface features. Ng and Cardie (2002c) report that their algorithm has only moderate performance, especially precision, on common noun phrases (i.e., the most difficult anaphors, compared to much easier cases of pronouns and proper names). This leads us to the hypothesis that if we want to resolve more difficult cases, we have to incorporate deeper knowledge into our algorithm.

A variety of semantic constraints and preferences may help a coreference resolution system. Consider the following examples:

(54) [Schwartz] said [Monday] that there were [more than 3.9 billion people] in [the world] without [telephone service] and [30 million people] currently on [waiting lists]. If $[\text{Globalstar}]_1$ begins [[its] service] on [schedule] in [1998], [he] predicted that $[\text{the company}]_{2,ante=1}$ would have [3 million customers] by [2,002], bringing in [$2.7 billion] in [annual revenue].

(55) [The Clinton administration] recently decided not to sanction China over the sale to Pakistan of materials used to make enriched uranium, on the

grounds that [central Chinese authorities]$_{2,ante\neq1}$ say they did not know of the sale and now promised not to make such sales again.

(56) [The sound] occurred about [5 minutes and 47 seconds] after [takeoff]; [the captain] "questioned" [the sound]1; 17 seconds after hearing [it]$_{2,ante=1}$, she said the plane, which was still climbing on its way to Atlanta, must return to Miami.

In Example (54), there are numerous candidate antecedents for the anaphor "the company". However, most of them are semantically incompatible: for example, "the world" and "the company" cannot possibly refer to the same entity. The only appropriate candidate is "Globalstar" — a proper name which could be marked as ORGANIZATION by an NE-tagger.

In Example (55), we need more sophisticated inference to resolve "central Chinese authorities": *administration* and *authorities* alone are semantically compatible. We can consult a knowledge base to find out that "Clinton administration" is in fact U.S. authorities, that U.S. and China are two different countries, and that two different countries cannot share the same authorities. An alternative solution would be to analyze the snippet directly: if "central Chinese authorities" and "The Clinton administration" denote the same entity, then we might substitute one for another, coming up with the following situation: "Central Chinese authorities recently decided not to sanction China. . . ". It is very unlikely that any authorities impose sanctions on their own country or even make any decisions on this issue. Consequently, the anaphoric link between the two markables is very implausible. Both strategies, however, rely on rather complex reasoning with a very large knowledge base. Unfortunately, such knowledge bases are not readily available for different domains and languages.

Example (56) shows how semantic preferences may help for pronoun resolution. Although most of the candidates are semantically compatible with the pronoun "it", the preferred antecedent is "the sound", as it is the most probable object of the verb *hear*: obviously, "hearing the sound" is more likely than "hearing the takeoff" or "hearing the captain", although the both latter cases are still possible.

In this study we only concentrate on semantic compatibility: for a given pair of markables, we try to automatically determine if they potentially can denote the same entity.

The extraction of semantic knowledge is one of the most difficult subparts of a coreference resolution algorithm. Semantic properties of markables are not explicitly present in the document, and, thus, one should consult an external information source, for example, the WordNet ontology (Miller, 1990). This raises two problems: some words, in particular, proper names, have no WordNet entries, and many others have more than one.

Using a Word Sense Disambiguation (WSD) algorithm could resolve these problems at least in part. However, this issue is outside the scope of the thesis for two reasons. First, to investigate the importance of WSD for coreference resolution, we need a corpus manually annotated with all the relevant information (i.e., word senses and coreference at the same time). Second, we aim at a less resource-intensive approach that can potentially be ported to other domains or even languages. Most WSD systems, however, require a lot of new training material to be adjusted to another domain.

Even if we have the correct lexical information for the markables themselves, we need efficient procedures to determine if their semantic properties are compatible: as we discuss below, this is a non-trivial task.

The next section briefly summarizes relevant studies in the field. In Sections 5.2–5.4 we investigate different strategies to account for semantic compatibility. Our experiments (Section 5.5) empirically assess the utility of semantic knowledge for nominal anaphora and for full-scale coreference.

5.1 Related Work

Early approaches to coreference resolution, especially pronominal anaphora (for example, (Wilks, 1973)) relied extensively on semantic and world knowledge. This information was encoded manually, limiting the flexibility of such approaches.

It was soon observed that creating an extensive general-purpose set of semantic constraints manually might not only be time consuming, but also practically infeasible. Consider an example from Carbonell and Brown (1988):

(57) a. John took the cake from the table and ate it.
b. John took the cake from the table and washed it.

We need the following semantic knowledge to resolve "it" in the first sentence: "*cake* is edible", "*table* is not edible", and "the object of *eat* is edible". It can be theoretically inferred from the dictionary entries of *cake* and *table* — *cake* is some kind of *food*, and *table* is not. The second sentence looks similar, but is in fact much more complex. The following information is required here: "*cake* is not washable", "*table* is washable", and "the object of *wash* is washable". In this case the dictionary cannot help us much — it is not true that *food* is generally not washable, and also introducing any subconcepts for washable vs. not washable food does not make any sense. Consequently, to resolve (57b) we either have to encode all the verb-object combinations directly, or rely on a very sophisticated inference machinery.

Since rule-based full-scale semantic processing is hardly feasible, several other approaches to anaphora resolution have been developed: limited do-

main applications (Hayes, 1981), knowledge-poor algorithms (Kennedy and Boguraev, 1996), or approaches relying on existing resources, ontologies and corpora, to automatically extract semantic constraints and preferences (see below). The main semantic resource for most existing coreference resolution algorithms is the WordNet ontology. Although, as we have mentioned above, coverage and ambiguity are important issues in this case, most approaches do not address these problems and simply map each markable to the first sense of its head noun (or the UNKNOWN tag in out-of-vocabulary cases).

Having mapped individual markables to ontology concepts, one faces another problem: accounting for semantic consistency for coreference chains. Most approaches, for example, Soon et al. (2001), rely on semantic class agreement — for each markable, a superconcept of a predefined granularity (for example, "LOCATION") is computed; markables having the same superconcepts are considered semantically compatible and the corresponding coreference links — semantically consistent. We discuss semantic class agreement in Section 5.2.

Harabagiu et al. (2001) argue that this view is too simplistic and propose a mechanism for mining "patterns of semantic consistency" — specific subgraphs of the WordNet ontology linking head nouns of coreferring descriptions. This approach is described in Section 5.4.

Semantic consistency is connected to similarity — usually quantified by measuring the relatedness or closeness of two concepts. Roughly speaking, similar concepts tend to be semantically compatible and vice versa. Numerous measures of semantic similarity, relying on WordNet and corpus counts, have been proposed in the literature (Resnik, 1995; Jiang and Conrath, 1997; Hirst and St-Onge, 1998; Leacock and Chodorow, 1998; Lin, 1998). Nevertheless, we are not aware of any approach incorporating WordNet similarity into a coreference resolution engine. We discuss the interaction between semantic similarity and coreference in Section 5.3.

The use of an existing general-purpose dictionary for coreference resolution involves a range of design choices, from mapping markables to predefined concepts to extracting the desired semantic constraints and preferences from the ontology structure. This may affect the resolution quality. Consequently, alternative ways of acquiring relevant semantic knowledge have been proposed in the literature. Dagan and Itai (1990) rely on corpus cooccurence counts to resolve the pronoun "it". Applied to our example (57), their system would collect corpus evidence for *cake* and *table* being objects of *eat* and *wash* and make respective choices. Simulating this parser-dependent approach with Internet counts, we get the following results for the Google search engine: "eat * cake" — 74500 counts, "eat * table" — 6740, "wash * cake" — 633, and "wash * table" — 3650. These data clearly suggest a preference for "it"="the cake" for (57a) and "it"="the table" for (57b).

Poesio et al. (1998), followed by Poesio et al. (2002), identify several problems with WordNet as a knowledge source for bridging anaphora resolution

and propose a vector-based method for unsupervised lexical acquisition augmented with syntactic patterns for meronymy. This approach is especially relevant for languages other than English, where electronic dictionaries are scarce (Gasperin and Vieira, 2004).

Finally, Bunescu (2003) presents a web-based algorithm for resolving definite description: using the Internet cooccurence statistics, antecedents for both identity and bridging anaphora are recovered. A similar approach has been proposed by Modjeska et al. (2003) for "other" anaphora.

To summarize, semantics-oriented approaches have been proposed in the literature from the early stages of research on coreference resolution. These algorithms, however, suffer from the lack of extensive bases of common-sense and lexical knowledge: even the richest ontologies, such as WordNet, have limited coverage and require sophisticated inference procedures to extract semantic constraints and preferences from their structure. These problems are still far from being resolved. Consequently, although it is generally admitted that semantic knowledge is crucial for coreference resolution, most state-of-the-art approaches still rely only on very simple properties, such as gender and semantic class agreement.

5.2 Semantic Class and Gender Agreement

In the present and two following sections we focus on comparing semantic properties of markables, thus, incorporating WordNet information into our coreference resolution system. We start with semantic class agreement for different granularity levels (present Section), proceed to similarity metrics reflecting both WordNet and corpus-related properties of markables (Section 5.3) and finally investigate more complex WordNet graph-based features (Section 5.4).

Coreferring descriptions denote the same entity, therefore, we might conclude that they should have common semantic properties. As corpus data show, this is not always true — an anaphor and its antecedent often describe the same object from different perspectives:

(58) Telepiu SpA and Cecchi Gori Group unveiled a [nine-channel package]$_1$ of digital pay-television programming for Italy and said they will sign up four more channels in the next few weeks.
...
"[This]$_{2,ante=1}$ is a [revolution]$_{3,ante=2}$ in television," he said. "And it's great for Italy that it is leading the way in the digital era."
...
Subscribers to the [new service]$_{4,ante=3}$ will have to buy a total package of satellite dish, decoder and a smart card.

Obviously, *package*, *revolution*, and *service* are not synonymous, but still are annotated as referring to the same object[1]. Even similar descriptions can be labelled differently following the WordNet principles:

(59) Mike McNulty, the FAA air traffic manager at Amarillo International, said the previous aircraft count, conducted in late 1994, was a "manual count on a pad," done informally by air traffic controllers. That 60-day accounting estimated that 25 planes a day entered flight patterns over [the zone]$_1$ where plutonium is stored in concrete bunkers covered with earth and grass.
...
Two of the airport's dozen flight paths are directly above [the nuclear storage area]$_{2,ante=1}$, McNulty said.

Figure 5.1 shows a part of the WordNet ontology covering all the markables in this snippet. As we see, *zone* and *area* are not synonyms and not even hyponyms. They show, however, semantic class agreement, when the classes are defined coarsely.

This example illustrates a clear precision/recall trade-off arising from the underlying classification: if we have only few classes (LOCATION vs. OBJECT), many non-coreferring markables show semantic agreement ("a pad", "plutonium", "25 planes", "the airport", and "concrete bunkers" all fall into the OBJECT class), whereas if the classification is too fine-grained (AREA vs. GEOGRAPHICAL_AREA), even coreferring NPs may show disagreement ("the nuclear storage area" and "the zone" fall into different classes). In this section we investigate different semantic classifications, and compare them with respect to how they interact with coreference.

Each classification scheme corresponds to a sub-forest of the WordNet hierarchy. This allows us to define semantic class agreement in two ways:

"same classes": Two markables belong to the same semantic class if the WordNet labels of (the first sense) of their head nouns are exactly the same. For example, "a pad" and "a pad" share the same class for all schemes, whereas "a pad" and "it" (as well as "a pad" and "plutonium") only for very coarse-grained classifications.

"compatible classes": Two markables belong to compatible semantic classes if the WordNet labels of (the first sense) of their head nouns are in a hyponymy/hyperonymy relation. For example, "a pad" and "it" have compatible semantic classes for any underlying scheme.

[1]The MUC-7 guidelines do not provide any explicit instructions for annotating such examples. An accurate analysis of this snippet involves a rather complex model of "event anaphora" (for "This" and "revolution"), investigated, for example, by Webber (1979).

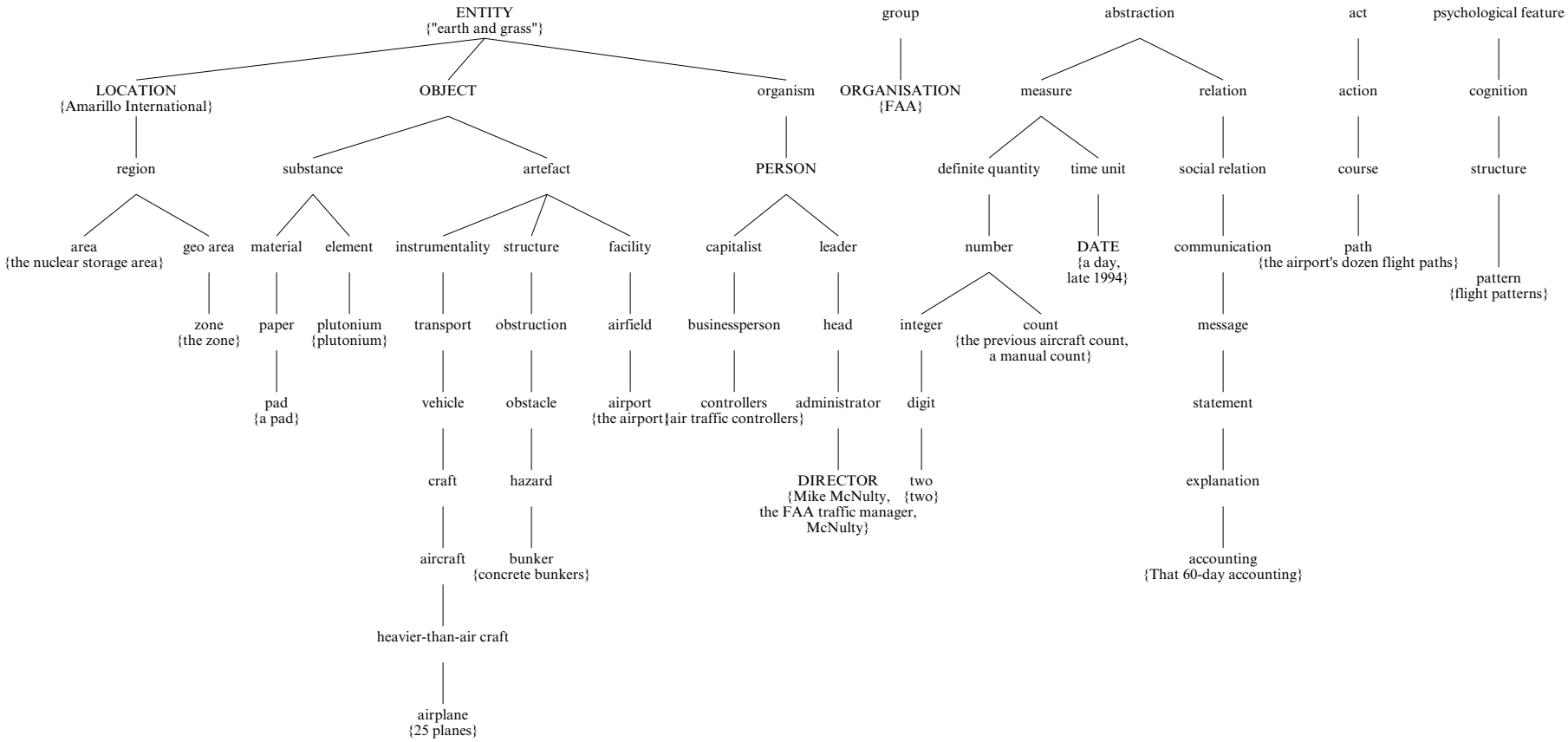


Figure 5.1: Fragment of the WordNet hierarchy for Example (59), markables shown in curly brackets, NE classes capitalized.

Below we show the corpus statistics for both definitions of agreement and for different granularity levels. In all the cases investigated, the distribution of ±coreferent links for +same (+compatible) pairs is significantly different from the distribution for −same (−compatible) pairs (χ^2-test, $p < 0.01$). We start with the most fine-grain WordNet-based classification and proceed to coarser-grained schemes.

WordNet-based classification. According to this very fine-grained classification scheme, each markable is mapped to the first WordNet synset of its head word. If the word is a pronoun, we map it to OBJECT, PERSON, FEMALE, or MALE. If the markable is a Named Entity, we use the labels obtained from our fine-grained NE classification algorithm (for PERSON and LOCATION, see below) or C&C NE tags (for other NE types). For coordinate constructions, we take the lowest superordinate (the concept dominating all the parts of a coordination).

Table 5.1 shows the distribution of ±coreferent links for pairs with ±same or ±compatible WordNet classes. We see that markables with the same WordNet labels more often tend to be coreferent compared to pairs with different WordNet labels (19% against 1%). Unfortunately, only few coreferent pairs share the same label (36.2%, 2547 of 7039 for the training data).

For compatible classes, we lose some precision: only 11% of compatible pairs in the training data are indeed +coreferent compared to 19% for same labelled pairs. However, more coreferent markables have compatible labels: 45.3%, 3187 of 7039.

The same tendency is stronger for common NPs. For proper names, however, we see that the precision goes down: only 14% of both +same and +compatible pairs are indeed coreferent. This can be explained by the fact that we assign non-specific WordNet labels to named entities. For example, "late 1994" in the snippet above is classified as DATE, together with "02-14", "later this year", "May 15", "14 months", "late summer", and "02-14-96" from the same document.

NE-motivated classification. The WordNet-based classification is very fine-grained for common nouns, but, in the same time, much coarser for named entities. To deal with this asymmetry, we have investigated another classification scheme, mirroring the granularity level of our fine-grained NE subclassification module, described in Uryupina (2005). This results in a much simpler data-driven semantic classification, shown on Figure 5.2.

This classification is a subtree of the original WordNet ontology, with all the upper concepts except ENTITY linked to the OBJECT node[2]. Therefore,

[2]This was done to ensure compatibility with Soon et al.'s (2001) classification scheme described below.

WordNet Class	Training data +coreferent		−coreferent	
All pairs				
+same	2547	19%	10903	81%
−same	4492	1%	350918	99%
+compatible	3187	11%	24586	89%
−compatible	3852	1%	337235	99%
Pairs of two common nouns				
+same	589	33%	1205	67%
−same	568	0%	122668	100%
+compatible	733	15%	4235	85%
−compatible	424	0%	119638	100%
Pairs of two NEs				
+same	1065	14%	6507	86%
−same	371	1%	33515	99%
+compatible	1117	14%	6979	86%
−compatible	319	1%	33043	99%

Table 5.1: Distribution of ±coreferent links for pairs with ±same and ±compatible WordNet classes in the training data (30 MUC-7 "dry-run" documents).

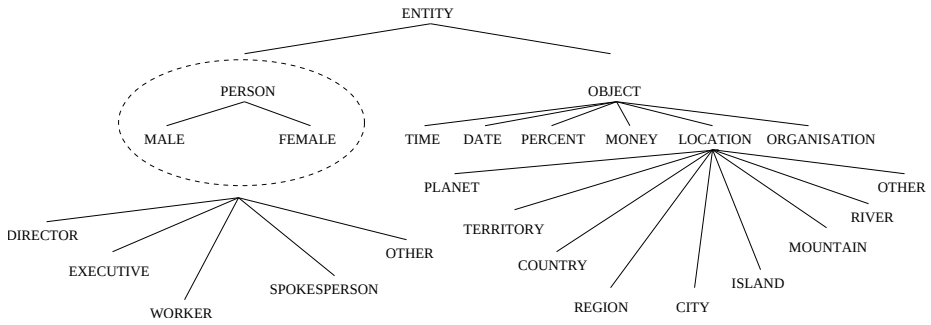


Figure 5.2: Semantic classes: NE-motivated scheme (Uryupina, 2005).

Figure 5.3: Semantic classes: scheme proposed by Soon et al. (2001).

to compute a label for a markable, we first obtain its class according to the WordNet-based scheme presented above, and then go up the WordNet tree until we find the corresponding label. To account for pronouns, we have supplemented the scheme with two gender classes, FEMALE and MALE, compatible with all the subclasses of PERSON (that is, DIRECTOR, EXECUTIVE, SPOKESPERSON, WORKER, and PERSON_OTHER).

For NEs, we rely on a web-based bootstrapping module (Uryupina, 2005) to automatically obtain the labelling. The algorithm is an extension of our earlier work (Uryupina, 2003b; Ourioupina, 2002) on automatic extraction of lexical knowledge from the Internet data. It combines automatically induced gazetteers, syntactic heuristics, and context features of Fleischman and Hovy (2002) to subclassify names of LOCATIONs and PERSONs into the classes shown on Figure 5.2. We point the reader to the mentioned papers for more details.

Table 5.2 shows the distribution of ±coreferent links for pairs with ±same or ±compatible NE-motivated classes. Predictably, we observe much more +same or +compatible pairs for this classification scheme than we have seen above (Table 5.1). This results in fewer cases of disagreement between coreferring markables: 3298 (of 7039, 46.9%) for +same and 5489 (78.0%) for +compatible compared to 2547 (36.2%) and 3187 (45.3%) in Table 5.1. The trend is even more visible for common NPs. In the same time, we see a big precision drop: only 4% of +same or +compatible pairs are indeed coreferent.

Soon et al's (2001) classification. All Coreference Resolution systems we are aware of, use similar coarse-grained classification schemes. We have chosen the ontology adopted by Soon et al. (2001) as an example. Compared to the NE-motivated classification presented above, this is essentially the same scheme, but without finer-grained distinctions for PERSON and LOCATION. It is shown on Figure 5.3.

The labels are computed in the same way as above: we first obtain WordNet information and then go up the tree to find an appropriate label.

Table 5.3 shows the distribution of ±coreferent links for pairs with ±same or ±compatible Soon et al.'s (2001) classes. For common NPs, this scheme works better than more fine-grained NE-motivated classification investigated above: with the same precision level, more coreferent links are considered +same or +compatible (1019 (of 1157, 88.1%) and 1028 (88.9%) compared to 936 (80.9%) and 946 (81.8%) respectively).

For proper names, this scheme over-relates slightly more than the NE-motivated classification: only 13% (compared to 16% in Table 5.2) of +same pairs are indeed coreferential.

WordNet Class	Training data +coreferent		−coreferent	
All pairs				
+same	3298	4%	83184	96%
−same	3741	1%	278637	99%
+compatible	5489	4%	117205	96%
−compatible	1550	1%	244616	99%
Pairs of two common nouns				
+same	936	2%	55588	98%
−same	221	0%	68285	100%
+compatible	946	2%	56876	98%
−compatible	211	0%	66997	100%
Pairs of two NEs				
+same	999	16%	5260	84%
−same	437	1%	34762	99%
+compatible	1322	15%	7431	85%
−compatible	114	0%	32591	100%

Table 5.2: Distribution of ±coreferent links for pairs with ±same and ±compatible NE-motivated classes in the training data (30 MUC-7 "dry-run" documents).

WordNet Class	Training data +coreferent		−coreferent	
All pairs				
+same	3792	4%	91213	96%
−same	3247	1%	270608	99%
+compatible	5753	5%	105754	95%
−compatible	1286	0%	256067	100%
Pairs of two common nouns				
+same	1019	2%	57380	98%
−same	138	0%	66493	100%
+compatible	1028	2%	57522	98%
−compatible	129	0%	66351	100%
Pairs of two NEs				
+same	1070	13%	7110	87%
−same	366	1%	32912	99%
+compatible	1337	16%	7156	84%
−compatible	99	0%	32866	100%

Table 5.3: Distribution of ±coreferent links for pairs with ±same and ±compatible Soon et al.'s (2001) classes in the training data (30 MUC-7 "dry-run" documents).

Gender agreement (English) gender can be seen as a classification scheme with the following classes: MASCULINE, FEMININE, and NEUTER. In addition, some nouns (PERSON label) are ambiguous between MASCULINE and FEMININE, and some (ANY label) are not marked for gender at all. This is the most coarse-grained classification we have investigated, so, we expect this kind of agreement to be a contra-coreference predictor — if two markables disagree in gender, they probably do not corefer.

For common nouns, the gender is computed by climbing up the WordNet tree until we find one of the appropriate labels. If none is found, the default label OBJECT is assigned. For pronouns, we consult a precompiled list. For Named entities, in particular PERSONs, we cannot proceed this way: our fine-grained NE classification module relies on a classification that does not reflect the MASCULINE vs. FEMININE distinction. We therefore have designed an alternative procedure for determining the gender of proper names automatically.

We have downloaded lists of female (2173 items) and male (1644) first names from the Web[3]. To classify a proper name, we first check whether it contains a gender descriptor (*Mrs.*, *Mr.*,...). If no descriptor can be found, we extract the first proper noun (tagged NNP) in the name and look it up in the list. If the noun can be found in neither FEMININE, nor MASCULINE list, we assign the PERSON tag. This procedure obviously has several shortcomings: first, it is not appropriate to foreign names of non-European structure (see examples in Section 3.2). Second, as any automated approach, it is error prone.

We have manually annotated our validation data with the gender information to evaluate the performance of the gender assignment module. The following tags were used for the manual annotation: FEM ("Hillary Clinton"), MASC ("Bill Clinton"), PERS ("Clinton"), OBJ ("White House"), and ANY ("They").

Table 5.4 shows the confusion matrix for our gender module together with its precision, recall, and F-score. The following two kinds of mistakes are the most frequent: MASCULINE nouns classified as PERSON and PERSON nouns classified as OBJECT. The main reason for the misclassification of MASCULINE names is the way this information is encoded in WordNet: in many cases, to preserve the tree structure, gender is not shown explicitly. For example, the unambiguously MASCULINE noun *father* is not a descendant of the *male_person* node, but is linked directly to *parent* instead. Most of PERSON markables misclassified as OBJECT are coordinate constructions, often containing a parsing mistake — these markables are placed too high in the hierarchy and are classified as OBJECT by default.

Table 5.5 shows the distribution of ±coreferent links for pairs with ±same

[3]www.pleasantcrab.com/gundel/names.tru

Annotated	Automatically assigned classes						Performance		
classes	F	M	P	OBJ	ANY	-	Recall	Precison	F-score
FEM	23	1	3	0	0	1	82.1%	92.0%	86.8%
MASC	2	56	17	3	0	0	71.8%	86.2%	78.3%
PERS	0	8	151	25	0	4	80.3%	84.8%	82.5%
OBJ	0	0	6	511	0	4	98.1%	94.1%	96.1%
ANY	0	0	1	4	24	6	68.6%	100%	81.4%

Table 5.4: Performance of the gender assigning module on the validation data (3 MUC-7 "formal train" documents).

or ±compatible gender classes. We see that ±compatible gender is indeed a good contra-coreference predictor: only 1% of −compatible pairs are coreferential. However, 9.5% (668 of 7039) of +coreferent pairs have incompatible gender. This is a result, on the one hand, of mistakes in gender assignment, and, on the other hand, of true disagreement, as, for example, between "We" and "the company" — an OBJECT noun can be used metonymically to denote a group of PERSONs. Still, we see fewer [−compatible, +coreferent] pairs for gender than for any other scheme.

In sum, semantic class agreement provides valuable information for coreference resolution. With fine-grained classification schemes, agreement is an indicator for coreference, whereas with coarse-grained schemes, disagreement is an indicator against coreference. However, none of the investigated schemes alone is sufficient to fully model semantic compatibility. In the following sections we will discuss more sophisticated ways of comparing NPs' semantic properties.

5.3 Semantic Similarity

In the previous section we have presented a way of estimating, whether two NPs are semantically close: based on a predefined set of WordNet-related classes, we compute a yes/no value. Our example (59) and the corresponding Figure 5.1 show a clear precision/recall trade-off. Fine-grained classification schemes inevitably assign different labels to very similar markables (such as "the plutonium storage area" and "the zone"). Coarse-grained schemes, on the contrary, group very different descriptions into the same class ("a pad" and "plutonium"). To deal with this problem we shift from our boolean agreement functions to continuous similarity metrics.

Several WordNet similarity measures have been proposed in the literature. They have been tested for a variety of NLP tasks, for example, Word Sense Disambiguation (Agirre and Rigau, 1996) or spelling correction (Budanitsky and Hirst, 2001). To our knowledge, there have been no attempts so far to

WordNet Class	Training data +coreferent		−coreferent	
All pairs				
+same	4840	2%	236627	98%
−same	2199	2%	125023	98%
+compatible	6371	2%	252960	98%
−compatible	668	1%	108690	99%
Pairs of two common nouns				
+same	1052	1%	89782	99%
−same	105	0%	33980	100%
+compatible	1094	1%	90237	99%
−compatible	63	0%	33525	100%
Pairs of two NEs				
+same	1283	4%	30261	96%
−same	153	2%	9761	98%
+compatible	1338	4%	30620	96%
−compatible	98	1%	9402	99%

Table 5.5: Distribution of ±coreferent links for pairs with ±same and ±compatible gender classes in the training data (30 MUC-7 "dry-run" documents).

apply WordNet similarity metrics to the Coreference Resolution task.

We have reimplemented four similarity measures[4] proposed by Resnik (1995), Jiang and Conrath (1997), Lin (1998), and Leacock and Chodorow (1998). We do not rely on already existing similarity packages (e.g., `www.d.umn.edu/~tpederse/similarity.html`), because our data, the MUC-7 corpus, mainly contain domain-specific texts. Below we briefly introduce these measures and compute their values for the $\{area,\ zone\}$ pair (see Figure 5.1 for the corresponding WordNet forest). For each metric, we compute the similarity value between the first synsets of the words and the maximum across the possible values for different pairs of synsets(minimum for $dist_{JC}$).

Leacock and Chodorow (1998) relate WordNet similarity to the tree-based distance between synsets syn_1 and syn_2:

$$sim_{LC}(syn_1, syn_2) = -\log \frac{path\_length(syn_1, syn_2)}{2D}.$$

The only semantic relation considered when computing the path between two synsets is hyponymy/hyperonymy, so, the path always goes from the synset syn_1 up to the lowest superordinate (in our example *region*) and then down

[4]Other popular approaches, for example, advocated by Lesk (1986) or Hirst and St-Onge (1998), are not considered here for reasons of computational complexity.

to the synset syn_2 The path length is then normalized by the overall depth D of the taxonomy. The similarity between "area" and "zone" in our Example (59) is $sym_{LC}(area, zone) = -\log \frac{4}{2D}$.

Resnik (1995) combines WordNet knowledge with corpus data, defining the similarity between two synsets as the information content of their lowest superordinate (lso):

$$sim_R(syn_1, syn_2) = -\log p(lso(syn_1, syn_2)),$$

where $p(syn)$ is the corpus probability of syn. Note that, unlike the other three similarity measures discussed here, this function does not assign the highest possible similarity value to pairs of the same synsets. If we consider the snippet (59) to be our whole corpus, $sim_R(area, zone) = -\log(p(region)) = -\log \frac{2}{23}$ and $sim_R(Amarillo\ International,\ Amarillo\ International) = -\log \frac{3}{23}$.

Jiang and Conrath (1997) propose a formula for measuring semantic distance — the counterpart of similarity — by combining the information content of the lowest superordinate with the information contents of the individual synsets:

$$dist_{JC}(syn_1, syn_2) = 2\log p(lso(syn_1, syn_2)) - (\log(p(syn_1)) + \log(p(syn_2))).$$

For our example, $dist_{JC}(area, zone) = 2\log \frac{2}{23} - (\log \frac{1}{23} + \log \frac{1}{23})$.

Lin (1998) proposes a general theory of similarity between arbitrary objects, combining the same elements as Jiang and Conrath (1997) in a different way:

$$sim_L(syn_1, syn_2) = \frac{2\log p(lso(syn_1, syn_2))}{\log(p(syn_1)) + \log(p(syn_2))}.$$

For our example, $sim_L(area, zone) = \frac{2\log \frac{2}{23}}{\log \frac{1}{23} + \log \frac{1}{23}}$.

WordNet Similarity and coreference. We have computed similarity measures for coreferent and non-coreferent pairs in our corpus. To apply the χ^2-test, we have discretized (continuous) similarity values into 10 bins. For all the four similarity measures, we have observed statistically significant difference (χ^2-test, $p < 0.01$) between the similarity distributions for ±coreferent pairs.

Figure 5.4 shows the distribution of discretized similarity values (maximum over all the synset combinations) normalized by the total of ±coreference links — for example, 60% of all the +coreferent links have the $dist_{JC}$ value below 0.1 (first bin).

Generally, we see that semantically similar markables more often tend to be coreferent, than semantically distant pairs. A remarkable exception is observed with Resnik's (1995) measure: we see a lot of +coreferent pairs in the second and fourth bins. This reflects the peculiarity of Resnik's approach — it only relies on the lowest superordinate, without paying any attention to the synsets

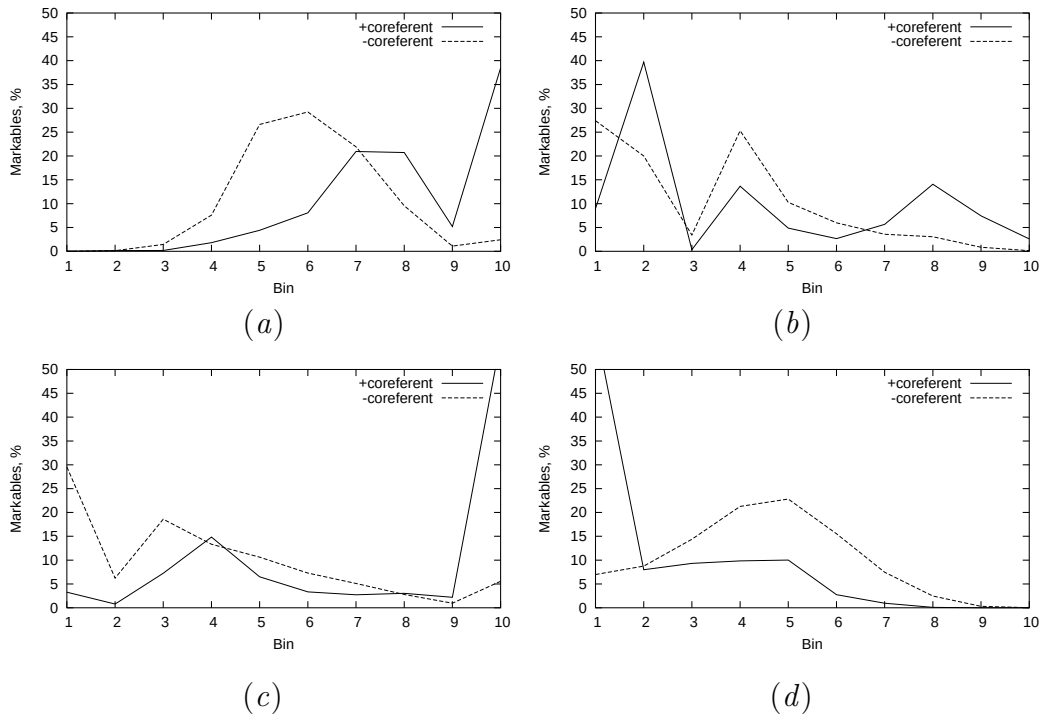


Figure 5.4: Discretized normalized WordNet similarity for ±coreferent pairs in the training data (30 MUC-7 "dry-run" documents), maximum (minimum for (d)) across all the senses: (a) Leacock and Chodorow (1998), (b) Resnik (1995), (c) Lin (1998), (d) Jiang and Conrath's (1997) distance.

themselves. Correspondingly, all pairs containing a pronoun are classified as highly non-similar (the second bin corresponds to the pronoun *it*, and the fourth bin to *he/she*): the lowest superordinate in such cases is the pronoun itself, which is very close to the top of the WordNet hierarchy and, thus, the probability of seeing it in the corpus is high.

5.4 WordNet Configurations

Semantic similarity measures are traditionally used for tasks such as Word Sense Disambiguation or Spelling Correction. Their goal is to cluster together words that are somehow related and, thus, often occur in the same text snippet. Resnik (1995) gives the example of *bicycle* and *car* — these two words should be considered similar.

We are, however, interested not in *similar* words, but in *compatible* ones — different descriptions of the same entity. Clearly, we do not want to con-

sider *bicycle* and *car* compatible, although they stay very close in the WordNet hierarchy. On the contrary, *package*, *revolution*, and *service* turn out to be compatible (see Example (58)), although not very similar. There is no straightforward way of extracting compatibility information: unlike similarity, these values cannot be obtained from WordNet directly.

Harabagiu et al. (2001) propose an algorithm for mining such knowledge from WordNet and a corpus annotated for coreference. They use the ontology structure to mine patterns of WordNet paths connecting pairs of coreferring nouns. As paths they consider sequences of the following relations: `synonymy`, `hyponymy`, `gloss`, `in-gloss`, `meronymy`, `morphology-derivation`, and `collide-senses`. For example, *beacon* and *signal* are connected via a `hyponymy:hyponymy` path: *beacon* is a *visual_signal* is a *signal*.

To obtain patterns of semantic consistency, Harabagiu et al. (2001) extract all the possible paths, compute their confidence values in an IR fashion and then rely on various ordering strategies to incorporate this information into their heuristics-based system. We adjust the idea of Harabagiu et al. (2001) for a machine learning set-up. The paths are encoded in a set of features and used to learn coreference rules. We therefore do not obtain any compatibility *function* — this information is only present implicitly.

We also adopt a simpler definition of a WordNet path: we only allow hyponymy links to be used. By adopting a more complex path definition, for example, the one proposed by Harabagiu et al. (2001), we get too many different path types with too few instances per type, and, consequently, a distribution that is too sparse for machine learning.

For an anaphor and an antecedent, we find their lowest superordinate (`lso`) in the WordNet hierarchy and then compute the path from the anaphor up to the superordinate and then back down to the antecedent (in our example the path between "the nuclear storage area" and "the zone" would be *area*→*region*←*geographical_area*←*zone*). We use the following information to encode paths as feature vectors: the WordNet ID of the lowest superordinate (`lso_id`), the distance between the lowest superordinate and the anaphor (`ana_lso_d`), the distance between the lowest superordinate and the antecedent (`ante_lso_d`), the depth of the lowest superordinate in the hierarchy (`lso_d`), the tuple {`ana_lso_d`, `ante_lso_d`}, the triples {`ana_lso_d`, `lso_id`, `ante_lso_d`} and {`ana_lso_d`, `lso_d`, `ante_lso_d`}.

These parameters are represented as a set of boolean features, for example `(lso=PERSON)` or `(ana_lso_d=2)&(ante_lso_d=1)`. For each function, we compute its values for all pairs of markables in the training data, for +coreferent pairs, and for −coreferent pairs. If the distribution for ±coreferent pairs differs significantly (χ^2-test, $p < 0.01$) from the overall distribution, the function is considered a positive/negative indicator. Too sparse indicators, affected less than 10 times, are discarded. Examples of negative and positive indicators are shown in Table 5.6.

Function	Example
Positive indicators	
`(ana_lso_d=1)&(ante_lso_d=2)`	(*area, zone*)
`(lso=PERSON)&(ana_lso_d=1)&(ante_lso_d=4)`	(*he, instructor*)
Negative indicators	
`(ana_lso_d=6)`	(*manager, airplane*)
`(lso_d=1)&(ana_lso_d=5)&(ante_lso_d=5)`	(*airport, broker*)

Table 5.6: WordNet configurations: examples of positive and negative indicators.

Table 5.7 shows the distribution of ±coreferent links for pairs with the path from the anaphor to the antecedent satisfying the conditions of positive, negative, and remaining indicators. We see a significant difference between the distribution for positive and negative indicators (χ^2-test, $p < 0.01$). This suggests that the extracted path configurations indeed contain information relevant for coreference resolution. Unfortunately, this approach suffers a lot from overfitting: the log-linear test shows a significant interaction ($p < 0.01$) between the indicator (positive, negative, or remaining), link (+coreferent or −coreferent), and corpus (training or validation) variables. In our experiments below we will see whether this issue affects the system's performance.

5.5 Experiments

Below we describe our evaluation experiments: with our semantic features (agreement, similarity and WordNet configurations), we train an SVM classifier and build a coreference resolution systems for common noun phrases (Experiment 5) and for all markables (Experiment 6). Our main source of semantic knowledge is the WordNet ontology, augmented with two external modules for proper names. The modules help assign fine-grained semantic labels (Uryupina, 2005) and determine markables' gender (Section 5.2).

Table 5.8 lists our WordNet-based features with the example values for the pair ("the nuclear storage area", "the zone").[5] To run the SVM^{light}learner, we have converted nominal values to boolean features — for each possible value, a separate feature is added. For configurations, only values corresponding to positive or negative indicators are considered. This results in 686 boolean and continuous semantic features.

[5]The similarity values $sim_R(area, zone)$, $dist_{JC}(area, zone)$, and $sim_L(area, zone)$ in Table 5.8 are different from those presented above: in Section 5.3, for illustrative purposes, we used a small toy corpus consisting of just a couple of sentences, whereas here we show the real values computed for the MUC-7 data.

WN-path indicators	Training data +coreferent		Training data −coreferent	
All pairs				
all pairs	7039	2%	361297	98%
Indicator type: `lso_id`				
positive	4691	13%	30768	87%
negative	2142	1%	325748	99%
rest	206	4%	4781	96%
Indicator type: `lso_d`				
positive	4853	10%	42149	90%
negative	2185	1%	319147	99%
rest	1	50%	1	50%
Indicator type: `ana_lso_d`				
positive	4368	10%	37502	90%
negative	2669	1%	323320	99%
rest	2	0%	475	100%
Indicator type: `ante_lso_d`				
positive	4754	7%	60106	93%
negative	2272	1%	299722	99%
rest	13	1%	1469	99%
Indicator type: {`ana_lso_d, ante_lso_d`}				
positive	5561	7%	69416	93%
negative	973	0%	264951	100%
rest	505	2%	26930	98%
Indicator type: {`ana_lso_d, lso_id, ante_lso_d`}				
positive	5405	11%	42288	89%
negative	612	0%	245434	100%
rest	1022	1%	73575	99%
Indicator type: {`ana_lso_d, lso_d, ante_lso_d`}				
positive	5515	11%	43012	89%
negative	648	0%	253763	100%
rest	876	1%	64522	99%

Table 5.7: Distribution of ±coreferent pairs for different WordNet configurations in the training data (30 MUC-7 "dry-run" documents).

Features	Range	Example value
Anaphor's parameters		
semclass_ne(M_i)	nominal	LOCATION
semclass_soon(M_i)	nominal	LOCATION
gender(M_i)	nominal	OBJECT
Antecedent's parameters		
semclass_ne(M_j)	nominal	LOCATION
semclass_soon(M_j)	nominal	LOCATION
gender(M_j)	nominal	OBJECT
Pair's parameters		
same_semclass_wordnet(M_i, M_j)	0,1	0
same_semclass_ne(M_i, M_j)	0,1	1
same_semclass_soon(M_i, M_j)	0,1	1
same_gender(M_i, M_j)	0,1	1
compatible_semclass_wordnet(M_i, M_j)	0,1	0
compatible_semclass_ne(M_i, M_j)	0,1	1
compatible_semclass_soon(M_i, M_j)	0,1	1
compatible_gender(M_i, M_j)	0,1	1
leacock_firstsense(M_i, M_j)	continuous	2.71
leacock_max(M_i, M_j)	continuous	2.71
resnik_firstsense(M_i, M_j)	continuous	4.56
resnik_max(M_i, M_j)	continuous	4.56
lin_firstsense(M_i, M_j)	continuous	0.67
lin_max(M_i, M_j)	continuous	0.67
jiang_firstsense(M_i, M_j)	continuous	4.48
jiang_max(M_i, M_j)	continuous	4.48
lso(M_i, M_j)	nominal	region
lso_d(M_i, M_j)	nominal	2
ana_lso_d(M_i, M_j)	nominal	1
ante_lso_d(M_i, M_j)	nominal	2
{ana_lso_d, ante_lso_d}(M_i, M_j)	nominal	{1, 2}
{ana_lso_d, ante_lso_d, lso}(M_i, M_j)	nominal	{1, 2, region}
{ana_lso_d, ante_lso_d, lso_d}(M_i, M_j)	nominal	{1, 2, 2}

Table 5.8: Semantics-based features and their values for the pair ("the nuclear storage area", "the zone") in Example (59).

5.5.1 Experiment 5: Using Semantic Knowledge to resolve common NPs

Semantic knowledge may help to resolve all kinds of markables. Thus, in Section 5.1 we have mentioned semantic-based approaches to pronoun resolution. Although name-matching is the main technique for resolving NEs, semantics may help in cases going beyond matching, for example, to distinguish between different entities with the same name (one of the documents in our corpus mentions three different "McDonald's"). For common NP resolution, however, semantic consistency is the most important information. In the present experiment we evaluate our WordNet-based approach on common NPs. We use the MUC-7 "dry-run" data to train an SVM classifier and test it on our validation set, the MUC-7 "formal training" corpus. All the anaphoric non-common NPs (i.e., pronouns and named entities) have been excluded from the annotation.

Baselines. Throughout this thesis, the algorithm of Soon et al. (2001) serves as the main baseline. This approach relies on just 12 features, called "basic" in our experiments. As a second, naive baseline, we have taken the "same surface" approach: two markables are considered coreferent, if they match after stripping off the determiners and lowercasing all the characters.[6] In all tables in this section we indicate significant improvements over the main baseline for $p < 0.05/p < 0.01$ by */** and significant losses — by †/††.

The first two rows of Table 5.9 show the performance of the baselines. The most surprising result is a very low precision level (52%) of the naive baseline: one could assume that two NPs having the same surface form should almost always be coreferent. To take a closer look at the problem, we have run our baselines on different subgroups of common NPs: the-NPs (introduced via the article "the"), definite NPs (introduced via the determiners "the", "this", and "these"), and other NPs (introduced via non-definite determiners — all except "the", "this", and "these").

The first two rows for each class of NPs show the baselines' performance. The precision for definite NPs, in particular the-NPs, is much higher than average. Other NPs, however, are problematic — even the very cautious "same surface" baseline achieves only moderate precision (37%).

Most approaches to coreference resolution either do not rely on any specific techniques for common noun phrases (Soon et al., 2001), or account exclusively for the-NPs (Grosz et al., 1983; Vieira and Poesio, 2000). Ng and Cardie (2002c) and Harabagiu et al. (2001) have designed mechanisms for common NPs resolution. Even these studies, however, do not provide a specific account for non-definites. Our corpus analysis (see Section 4.2) shows that up

[6] This corresponds to our `lowcase_nodeterminer_exactmatch` feature (Chapter 3).

to one third of all the anaphoric common NPs are introduced via a non-definite determiner. Therefore we have paid extra attention to this group, identifying the following problematic cases.

First, some non-definite NPs are parsing mistakes, such as "Japan ---". Second, several NPs, mostly singular nouns with no articles, are parts of multiword expressions, as "example" in "for example". Our same surface baseline merges all the "example" markables throughout a document into one chain. Parsing mistakes and parts of multiword expressions can be modeled by introducing an anaphoricity filter — before the main coreference engine starts, a preprocessing module would discard non-anaphoric markables. We will discuss this solution in Chapter 7.

The third class of mistakes are functional nouns, such as "lack". They are seldom used as standalone markables, but, instead, build constructions of the form "NP_1 of NP_2": "lack of clarification", or "lack of airplanes". In Section 4.5 we have introduced syntactic features to account for postmodification.

Finally, some misclassified anaphors, mostly, but not necessarily, bare plurals, are high-frequency nouns, either in general ("people", "something"), or for the MUC-7 domain ("airplanes"). They may appear at several places throughout the document, denoting different entities:

(60) "I'm so proud of [those people]$_1$ who fly," Wolcott said.
. . . (21 sentence)
Cheryl McNair confesses she is surprised by the level of concern [people]$_{2,ante\neq1}$ still express for her family 10 years after the accident.

The proximity information would help in this case. Thus, in (60), the skipped part in the middle spans over 21 sentence, making the link between "those people" and "people" very unlikely. More precise descriptions, however, often denote the same entity even far from each other in the document:

(61) But it was also a potentially deadly stunt, opposed by fire and safety officials, that required [a $400 million insurance policy]$_1$.
. . . (15 sentences)
The risk was considered such that the Host Committee took out [a $400 million insurance policy]$_{2,ante=1}$ for the helicopter stunt.

In this example, two mentions of "a $400 millions insurance policy" are coreferent, although the distance between them is 16 sentences. We will discuss proximity in Section 6.4

In sum, common nouns with non-definite determiners are intrinsically difficult. Most problematic cases, however, can possibly be solved by other modules, relying on syntactic, proximity, or anaphoricity knowledge.

Semantic features. The third and the fourth row for each markables' group in Table 5.9 shows the performance of semantic features. Improvements in recall come, unfortunately, with significant losses in precision. The figures get higher when we combine semantic knowledge with the feature set of Soon et al. (2001): compared to the main baseline, the precision goes up significantly (χ^2-test, $p < 0.01$), however the recall goes down (not significant, but still 22.2% relative deterioration), providing an F-score of 46.6% – 2.6% relative improvement over the baseline.

The same trend is observed for the subgroups of common NPs with the only exception of semantic features for definites, where both recall and precision deteriorate. For definite noun phrases, especially the-NPs, semantic knowledge alone and even in combination with Soon et al.'s (2001) features cannot outperform the baseline. For non-definites, semantics brings a slight performance gain.

To summarize, we see that WordNet-based features over-relate too much, considering too many markables as compatible. Although they help resolve NPs not covered by the baseline, their precision is not satisfactory and the overall benefit is only moderate.

Feature groups. To see, how different features contribute to the overall performance, we have conducted the same experiment for each feature group separately. The results are reported in Table 5.10. For every group of markables, the similarity-based classifier shows the worst performance (F-score), followed by agreement, followed by the configurations.

Combined with the feature set of Soon et al. (2001), the agreement and similarity features do not affect performance: the resulting classifier is almost the same as the one build upon the baseline features alone. Agreement brings not too much new information, because two of eight features, `compatible_gender` and `compatible_semclass_soon`, are already included in the baseline. Similarity features do not make any impact on the combined system either: they alone are poor predictors of coreference and therefore the machine learner relies more on the baseline features.

WordNet configurations perform better: overall, configurations alone show a higher F-score than all the semantic features together. They also change the behavior of the baseline: augmented with configurations, the system of Soon et al. (2001) shows higher recall and lower precision (significant differences for each markables' group are shown in Table 5.10). We may conclude that this feature group causes the over-relating problem we have observed above.

To summarize, our experiment shows that WordNet similarity is not extremely relevant for coreference resolution. The contribution of agreement cannot be evaluated directly in this set-up, because two of eight features are included in the baseline as well. Compared to the naive same surface baseline,

Algorithm	Validation set		
	Recall	Precision	F-measure
All common NPs			
"same NP" baseline	33.0	52.1	40.4
Soon et al. (2001), SVM baseline	39.3	50.0	44.0
semantic features	51.8	†35.6	42.2
basic and semantic features	**58.0	38.9	46.6
NPs with definite determiners			
"same NP" baseline	40.9	61.4	49.1
Soon et al. (2001), SVM baseline	42.4	62.2	50.4
semantic features	57.6	40.9	†47.8
basic and semantic features	57.6	†41.3	48.1
The-NPs			
"same NP" baseline	54.0	69.4	60.7
Soon et al. (2001), SVM baseline	58.7	68.5	63.2
semantic features	52.4	†47.8	50.0
basic and semantic features	65.1	53.3	58.6
Other NPs			
"same NP" baseline	21.7	37.0	27.4
Soon et al. (2001), SVM baseline	34.8	37.2	36.0
semantic features	50.0	29.9	37.4
basic and semantic features	50.0	30.3	37.7

Table 5.9: Semantics-based approach to common NP coreference, performance on the validation (3 MUC-7 "train" documents) data. Significant improvements over the "same NP" baselines are shown by */** and significant losses — by †/††($p < 0.05/p < 0.01$).

Feature Groups	Validation set Recall	Precision	F
All common NPs			
Soon et al. (2001), SVM baseline	39.3	50.0	44.0
agreement	51.8	†36.0	42.5
similarity	49.1	†29.0	36.4
WordNet configurations	51.8	37.2	43.3
all semantic features	51.8	†35.6	42.2
agreement+Soon et al. (2001)	39.3	53.0	45.1
similarity+Soon et al. (2001)	39.3	53.0	45.1
configurations+Soon et al. (2001)	**57.1	39.5	46.7
all sem. features+Soon et al. (2001)	**58.0	38.9	46.6

Table 5.10: Semantics-based approach to common NP coreference, performance for different feature groups on the validation (3 MUC-7 "train" documents) data. Significant improvements over the baseline (SVM^{light} with features of Soon et al. (2001) are shown by */** and significant losses — by †/††($p < 0.05/p < 0.01$).

agreement-based classifier provides a slight performance gain (42.5% against 40.4%). Wordnet configurations encode valuable information but over-relate too much. This reflects the overfitting problem we have observed in Section 5.4, analyzing the distributions for the training and validation data. A possible remedy would be an external corpus for mining configurations. Such a resource, however, could be probably better used as a supplement to our main training dataset.

5.5.2 Experiment 6: Semantics-based Full-scale NP-coreference Resolution

In this experiment we evaluate semantic features for full-scale coreference resolution. We use the MUC-7 "dry-run" data to train an SVM classifier and test it on the MUC-7 "formal testing" corpus. Throughout this thesis, we compare our results to the following two approaches: merging all markables into one chain (naive baseline) and an SVM classifier with Soon et al.'s (2001) features (our main baseline). In all tables in this section we show significant improvements over the main baseline for $p < 0.05/p < 0.01$ by */** and significant losses — by †/††.

Pure semantic approach. Table 5.12 shows the performance level of our semantics-based classifier compared to the baselines and syntax and matching-based approaches presented in Chapters 3 and 4.

Features	Test set Recall	Prec.	F	Validation set Recall	Prec.	F
Baselines						
"merge all"	86.6	35.2	50.0	91.9	38.0	53.7
basic features (Soon et al., 2001)	50.5	75.3	60.4	54.5	56.9	55.7
Semantic features						
agreement+ similarity+config.	††28.5	††48.3	35.9	††37.9	†48.5	42.6
Other knowledge types						
matching	52.2	††61.2	56.3	56.2	53.3	54.7
syntax	††7.6	68.5	13.8	††9.9	57.4	16.9

Table 5.11: A semantics-based approach to the full-scale Coreference Resolution task: performance on the testing (20 MUC-7 "formal test" documents) and the validation (3 MUC-7 "train" documents) data. Significant improvements over the baseline (SVM^{light} with features of Soon et al. (2001) are shown by */** and significant losses — by †/††($p < 0.05/p < 0.01$).

We see that both the precision and recall figures are significantly lower than the baseline (χ^2-test, $p < 0.01$). The precision drop is not surprising — we have already encountered this problem in Experiment 6. The recall drop suggests that, although semantic features help account for non-trivial cases of common NP coreference, for full-scale resolution semantics alone is not sufficient.

Compared to other knowledge sources, the pure semantics-based approach shows a moderate performance. It cannot compete with the shallow matching approach neither in precision nor in recall (both significantly worse, χ^2-test, $p < 0.01$). Compared to syntax, it shows a better F-score (35.9 against 13.8) as a result, however, of bringing closer precision and recall values.

To summarize, the pure semantic classifier over-relates too much and, as a result, suffers from low precision. Although the recall figures are higher for common NPs (Experiment 5), for the full-scale task they achieve only a very moderate level. We may conclude that the relevance of WordNet-based processing for coreference resolution is questionable, supporting the position of Poesio et al. (1998) and Poesio et al. (2002).

Combining Semantics with the Basic Coreference Features. Experiment 5 has shown that semantic knowledge alone is not sufficient for common NPs resolution, but, when combined with Soon et al.'s (2001) features, leads to a slight improvement. We have run the same experiment for full-scale coreference resolution.

Table 5.12 shows the performance figures of the semantic classifier aug-

Features	Test set Recall	Prec.	F	Validation set Recall	Prec.	F
Baselines						
"merge all"	86.6	35.2	50.0	91.9	38.0	53.7
basic features (Soon et al., 2001)	50.5	75.3	60.4	54.5	56.9	55.7
Semantics+basic features						
basic+agreement+ +similarity+config.	**55.6	††67.5	61.0	60.3	55.1	57.6
Other knowledge types						
basic+matching	**58.4	††63.1	60.6	60.6	54.1	57.1
basic+syntax	52.2	75.2	61.7	56.2	56.5	56.4

Table 5.12: Basic coreference resolution algorithm Soon et al. (2001) augmented with semantic knowledge, performance on the testing (20 MUC-7 "formal test" documents) and the validation (3 MUC-7 "train" documents) data. Significant improvements over the baseline (SVM^{light} with features of Soon et al. (2001) are shown by */** and significant losses — by †/††($p < 0.05/p < 0.01$).

mented with Soon et al.'s (2001) features. Similar to Experiment 5, the recall level increases, but precision drops (both significant for the test data, $p < 0.01$). This provides a slight gain in F-score.

In Chapter 8 we will combine our semantic features with other knowledge sources.

5.6 Summary

In this chapter we have investigated the influence of semantic features on coreference resolution: semantic class and gender agreement (Section 5.2), similarity (Section 5.3), and WordNet patterns of semantic consistency (Section 5.4).

The main source of semantic knowledge for our system is the WordNet ontology. However, it does not provide information for some markables, especially, proper names. To assign fine-grained semantic labels to NEs, we have developed a web-based bootstrapping approach (Uryupina, 2005).

To evaluate the impact of semantic knowledge, we have encoded it in 686 features, learned an SVM classifier and tested it on common NPs (Experiment 5, Section 5.5.1) and on all markables (Experiment 6, Section 5.5.2).

Both experiments show the same tendency: WordNet-based classifiers overrelate too much, showing low precision. The problem persists, although not to such extent, even when we combine semantic knowledge with the features of Soon et al. (2001). In Chapter 8 below we will incorporate semantic module

into our coreference resolution engine, combining it with with other knowledge sources.

Chapter 6

Discourse Structure and Salience

In the previous chapters we have investigated different information types that are potentially useful for Coreference Resolution: name matching, syntactic, and semantic evidence. In some cases, however, this knowledge does not give us enough evidence to pick the correct antecedent. Consider the following example:

(62) [Valujet Airlines Inc.]$_1$, which is based in [Atlanta]$_2$ and serves [26 cities]$_3$ in [17 states]$_4$, began [operations]$_5$ in [1993]$_6$ with [an old fleet]$_7$ that has grown to [about 50 planes$_8$]. [It]$_{9,ante=1}$ has experienced [a number]$_{10}$ of [problems]$_{11}$, including [several aircraft]$_{12}$ that have run off [runways]$_{13}$ and [one]$_{14}$ that caught [fire]$_{15}$, but never [a crash]$_{16}$.

In this snippet, we have 8 candidate antecedents for the pronoun "It". Half of them ("26 cities", "17 states", "operations", and "about 50 planes") can be filtered out by an agreement check. The remaining four markables ("Valujet Airlines Inc.", "Atlanta", "1993", and "an old fleet"), however, cannot be disambiguated neither by string matching nor by syntactic or semantic preferences. Moreover, agreement constraints, which correctly filter out erroneous antecedents in this example, can filter out correct antecedents in other cases:

(63) "[Satellites]$_1$ give [us]$_2$ [an opportunity]$_3$ to increase [the number]$_4$ of [customers]$_5$ [we]$_6$ are able to satisfy with [the McDonald's brand]$_7$," said [[McDonald's]$_8$ Chief Financial Officer]$_9$, [Jack Greenberg]$_{10}$. "[It]$_{11,ante=1}$ is [a tool]$_{12}$ in [[our]$_{13}$ overall convenience strategy]$_{14}$."

In (63), the 11th markable, "It", has 10 candidate antecedents. Syntactic constraints rule out four of them, including the correct markable "Satellites".

Humans obviously do not encounter any difficulties interpreting anaphors in these two snippets. This suggests that there exists another kind of information important for coreference resolution: after processing the first sentences of (62) or (63), "Valujet Airlines Inc." and "Satellites" are the most *salient* entities and thus are likely to be antecedents.

The idea of "salient entities" is intuitively very important for text understanding, but extremely difficult to formalize. Numerous theories of discourse and information structure try to account for this phenomenon, introducing notions such as *focus*, *topic*, or *center*. These frameworks have been developed as formal linguistic theories, so it is not always possible to straightforwardly use them in a computational project. For a fully automated approach, we need salience measures that, on the one hand, are robust enough to be reliably computed using real-world error-prone NLP modules, and, on the other hand, are theoretically sound and make predictions relevant for coreference resolution.

In the next section we briefly introduce the main principles underlying existing discourse theories and salience-based anaphora resolution algorithms. In Sections 6.2–6.7 we proceed to identify and evaluate different salience-affecting factors, ranging from very basic measures (for example, the markable's linear position in the document) to more elaborated ones (e.g., the backward-looking center or the candidates' coreferential properties). Section 6.8 introduces our experiments on salience-based pronoun resolution and full-scale coreference.

6.1 Related Work

Discourse modeling is essential for a variety of text-understanding problems, including anaphora resolution. Linguistic theories (Halliday and Hasan, 1976; Givon, 1983; McKeown, 1985; Reichman, 1985; Fox, 1987; Mann and Thompson, 1988; Kamp and Reyle, 1993; Grosz et al., 1995; Moser and Moore, 1996; Cristea et al., 1998) have been proposed in the past three decades to account for discourse structure and its relevance for the interpretation and generation of anaphoric expressions. It is impossible to give an extensive overview here, so we only highlight the most common principles. A list of seminal papers on various theories of discourse and information structure as well as a summary of current issues in the field can be found in (Kruijff-Korbayova and Steedman, 2003).

Texts are not arbitrary collections of sentences but, rather, their sentences are related to each according to some structure. This intuitive claim is the starting point of all the theories. It is also supported by recent empirical studies on sentence reordering (Karamanis, 2003; Lapata, 2003). Different scholars have proposed different representations of such structures and there is still no agreement among linguists on this issue. For example, RST structures (Mann and Thompson, 1988) are built with very specific relations ("evidence",

"cause",...), whereas Grosz and Sidner's intentional structures are formed only with very general relations, "dominance" and "satisfaction-precedence".

An important property of discourse structure, relevant for Coreference Resolution, is *coherence*. The most general claim about coherence is that discourses that keep mentioning the same entities are perceived as more coherent than those that do not (see, for example, "lexical cohesion" in (Halliday and Hasan, 1976)). Contemporary linguistic theories provide a more elaborated account of global and, especially, local coherence, relating it to *salience*. Intuitively, salience is a measure of entities' prominence at some point in the discourse. Different frameworks propose their own definitions of salience and factors affecting it.

A text segment[1] can be organized in a structure, an *information state*, reflecting the salience of its units. Some elements of this structure are considered the most important entities and play a crucial role in establishing (local) coherence. These elements are known in the literature under different names[2]: focus, topic, or center. These notions formalize very similar intuitions, but their properties still may vary for different theories. The exact definition of the main elements is an important research question addressed in the literature both theoretically and empirically (see (Poesio et al., 2004) for discussion).

The information state is a key concept in most theories. It reflects the entity-coherence of a text: in a coherent discourse, the information state is updated following some specific patterns. The position (for example, rank) of an entity in the information state determines its anaphoric accessibility — the likelihood of being an antecedent for some entity in subsequent discourse segments. Consequently, one can potentially use information states and their transitions for coreference resolution.

While linking discourse structure to anaphoric accessibility, linguistic theories make important statements for coreference resolution. However, many of these claims can hardly be tested empirically in a corpus-based study or used in a fully automated approach. For example, Marcu et al. (1999) reveal numerous difficulties encountered when annotating a corpus with the RST relations: although people generally agree on the boundaries for discourse segments, it is very hard to get agreement on relation labelling. Poesio et al. (2004) show that the accurateness and coverage of the Centering predictions depends crucially on the instantiations of various parameters.

Several symbolic pronoun resolution algorithms (Brennan et al., 1987; Tet-

[1]Here we try to stay clear from the particular theories and their terminology and therefore do not define text segments explicitly. Different approaches consider clauses, sentences, or bigger groups to be segments. We also try to use only the most general discourse-related concepts, for example, "information state" to avoid controversial terminology.

[2]We point the reader to the study of Kruijff-Korbayova and Steedman (2003) for the review of various relevant definitions in different theories. In Section 6.5 below we present the corresponding concepts of the Centering theory.

reault, 2001; Strube, 1998; Henschel et al., 2000) are based on discourse structure theories, especially on Centering. These approaches follow a similar scheme: after each utterance, the information state is updated by identifying and *ranking* all the entities according to some criterion. Following the underlying theory, each algorithm picks specific entities from the information state structure as candidate antecedents.

An alternative, more practically-oriented approach has been proposed by Lapin and Leass (1994) and further developed by Kennedy and Boguraev (1996). They identify basic factors affecting entities' salience and weight them to compute an overall salience value. On the one hand, this approach is more simplistic, compared to those based on discourse theories: salience values are computed independently for each entity and the structural information is lost. On the other hand, this approach allows to incorporate different salience-affecting factors and can be transformed into a scalable corpus-based algorithm (Preiss, 2001).

Salience has mainly been investigated for pronominal anaphora resolution. This is not surprising: (intersentential) pronominal coreference is an ideal testbed for salience-based approaches, because other information types are less relevant for pronouns than for full NPs. Theoretical studies, in particular (Sidner, 1979), however, suggest that discourse structure is an important factor for coreference resolution in general, not restricting its range to pronominal anaphors. Poesio (2003) has proposed a resolution procedure showing that salience is the most important factor for interpreting indirect NP-anaphora, outweighing, contrary to linguistic expectations, semantic information. Nevertheless, most approaches to full-scale anaphora resolution do not pay enough attention to salience.

Traditionally, most salience-based approaches to anaphora have been evaluated on application-specific small corpora. Recent studies show extensive evaluation of some of the centering algorithms (Tetreault, 2001) and claims (Poesio et al., 2004). Although addressing different tasks from different perspectives, these two studies come to the same conclusion: salience is a very important, but not the only factor in anaphora interpretation. This can be seen as an extra motivation for our approach — encoding salience in a group of features to combine it with other information sources in a machine learning set-up.

6.2 Document Structure

As we have seen in the previous Section, theoretical studies emphasize the importance of discourse knowledge for anaphora resolution. In the following Sections we will discuss relevant discourse and salience factors, encoding them as features for our machine learning experiments (Section 6.8). We start with

	Anaphors				Antecedents			
section tag	+anaphor		−anaphor		+antecedent		−antecedent	
SLUG	0	0%	71	100%	52	73%	19	27%
DATE	0	0%	30	100%	19	63%	11	37%
NWORDS	0	0%	30	100%	11	37%	19	63%
PREAMBLE	84	21%	314	79%	137	34%	261	66%
TEXT	1589	36%	2790	64%	1484	34%	2895	66%
TRAILER	30	50%	30	50%	0	0%	60	100%

Table 6.1: Distribution of anaphors vs. non-anaphors and antecedents vs. non-antecedents for different sections of a document in the training data (30 "dry-run" MUC-7 texts).

the basic document structure.

Coreference properties of an entity depend on its place in the document. First, our data come from the New York Times Newswire Service and already contain some SGML annotation, splitting each document into the DOCID, STORYID, SLUG, DATE, NWORDS, PREAMBLE, TEXT, and TRAILER sections[3] . Each section has its own structure that should be taken into account.

Second, the distribution of anaphoric vs. non-anaphoric markables depends on their position in the text: most entities get introduced in the beginning and then are further referred to throughout the document: around 50% of all the coreference chains in the training data and 30% in the validation data get started in the documents' headers or the first two paragraphs.

We encode the basic document structure in the `section tag` feature extracted from the SGML annotation (see Section 2.2 for an example of MUC-annotated data). According to the MUC scheme, the following parts of a document should be annotated for coreference: SLUG, DATE, NWORDS, PREAMBLE, TEXT, and TRAILER.

Table 6.1 shows the distributions of anaphors vs. non-anaphors and antecedents vs. non-antecedents for different parts of a document. For the training data, the `section tag` variable affects both the ±anaphor and ±antecedent distributions (χ^2-test, $p < 0.01$). For the validation data, we have merged SLUG, DATE, and NWORDS into one category and TEXT and TRAILER into another one to fulfill the assumptions of the χ^2-test. The resulting 3-valued `section tag` variable affects the ±anaphor (χ^2-test, $p < 0.01$) and ±antecedent (χ^2-test, $p < 0.05$) distributions.

Each part of the document has its own structure. For example, the SLUG

[3]The TEXT part is an article as it appears in NYT, all the other parts contain auxiliary information and follow the internal NYT guidelines. An example of SLUG is shown in 64 below.

section is a sequence of words separated with the hyphenation mark "-", starting with "BC", continuing with the most important keywords for the article, and optionally ending with a sequence of auxiliary quasi-words:

(64) <SLUG fv=taf-z> BC-LORAL-SPACE-470&ADD-N </SLUG>.

This structure follows several patterns specific for the NYT News Service that are not relevant for Coreference Resolution in general: knowing these patterns, we are able to extract better markables from the MUC data, but we obviously have to readjust this part of our algorithm for every new corpus. Keeping this in mind, we have mainly concentrated on the TEXT section: standard texts in English marked for paragraphs without any additional annotation or auxiliary elements.

The distribution of anaphors and antecedents in a text is affected by various discourse-level properties. As we have mentioned in the previous section, numerous linguistic theories investigate the role of discourse structure for anaphoric accessibility. Most studies, however, are mainly interested in the generation issue, especially, in pronominalization conditions: provided we have some material, how do we organize it to make a locally and globally coherent discourse? Consider the following discourses analyzed in various papers on Centering:

(65) a. John went to his favorite music store to buy a piano.
b. He had frequented the store for many years.
c. He was excited that he could finally buy a piano.
d. He arrived just as the store was closing for the day.

(66) a. John went to his favorite music store to buy a piano.
b. It was a store John frequented for many years.
c. He was excited that he could finally buy a piano.
d. It was closing just as John arrived.

These two examples present the same information about the same entities. However, they organize this information differently, choosing different forms of referring expressions. As predicted by the Centering theory, the first example is perceived as more coherent, concentrated around one topic (*John*).

In other words, linguistic theories are mainly focused on the interaction between the organization of written texts into sentences and paragraphs and the *form* of anaphoric expressions. In this section, on the contrary, we investigate the *distribution* of anaphors and antecedents depending on general discourse properties.

On both the global and local level, two main factors affect the distribution of ±anaphors or ± antecedents. On the one hand, for a text to be *coherent*, it should evolve around one topic, and, thus, contain a lot of anaphors, forming very few chains. The following examples show extreme cases of entity coherence:

(67) The higher $[\text{I}]_1$ climb, the hotter $[\text{I}]_{2,ante=1}$ gauge,
$[\text{I}]_{3,ante=2}$ cannot escape $[[\text{my}]_{4,ante=3}$ crystal cage$]_5$.
What am $[\text{I}]_{6,ante=4}$?

(68) $[\text{I}]_1$ can be crystal clear
Or dark as $[\text{pitch}]_2$.
$[\text{I}]_{3,ante=1}$ can be still and silent
Or $[\text{I}]_{4,ante=3}$ can rumble and roar.
What am $[\text{I}]_{5,ante=4}$?

Completely incoherent texts are normally perceived as at least strange and pointless, occasionally or on purpose:

(69) — Truly sport has become the universal language — no matter what tongue you speak. Make mine a pint of bitter and I'll explain how Glen Hoddle will run Spurs into Div. 1.....!
— Stern John will have GOTY, have the most Premiership goals, and Birmingham will still get relegated.
— We are Brummies! We are Brummies! Yes, we are! We are Brummies! Yes, we are!
— Oh how I miss pubs

This "pub talk" from an Internet chat can hardly be considered a uniform discourse: all the utterances are completely independent. Although such texts can be produced by humans, especially under the influence of drink or drugs, they fall beyond the scope of our present research.

On the other hand, for a text to be *informative*, it should address its topic from different sides, introducing more and more entities: despite such examples as (67, 68), a text of any reasonable length cannot normally refer to just one frequently repeated entity.

These two factors motivate the following naive scheme of a short discourse, supported by most documents in our corpus: the main topic and the corresponding entities get introduced in the beginning of the text, they start long coreference chains going all the way through the document and making it coherent. Locally, each discourse segment (paragraph) starts with some of these main entities and then brings in some new information, represented as short

coreference chains or singleton NPs. These new entities are related to the paragraph's topic and thus are not likely to be mentioned once again later.

Although the proposed hypotheses are too simplistic and cannot account for longer documents, they seem to work well for the short NYT articles (20-25 sentences). More importantly, they have several consequences that can be verified empirically:

1. Earlier segments in a document contain fewer anaphors than the later ones: entities get introduced in the beginning and only then are used as anaphors.

2. For the same reasons, earlier segments of a document contain more antecedents.

3. On the local level, entities in the beginning of a segment tend to be [+anaphor,+antecedent], whereas entities closer to the end of a paragraph tend to be −antecedent (but could be both ±anaphor): to relate a paragraph to the document's topic, it is normally started with some of the main entities, participating in long chains, and, thus, being anaphors and antecedents in the same time. New information is added later in the form of singleton NPs ([−anaphor,−antecedent]) or very short chains ([−anaphor,+antecedent] for the first element, [+anaphor,−antecedent] for the last one).

4. Short coreference chains are more likely to appear within a discourse segment than to span over more than one paragraph: unlike long chains, short ones usually correspond to local topics fully discussed within a single paragraph.

In this section we address the first three hypotheses, verifying them with the corpus data. The repetition-related hypothesis 4 is discussed in Section 6.7 below.

We use the original NYT annotation to compute the `paragraph number` for a given markable. We normalize these counts by the text's length in paragraphs. The resulting normalized counts are then discretized into 10 bins (for the 1st, 2nd,..., 10th part of the document) to avoid data sparseness. For each bin we compute the percentage of anaphors and antecedents among the corresponding markables.

Figure 6.1 shows the percentage of +anaphors and +antecedents for each bin in the training data. The first hypothesis is not supported by this analysis: the percentage of anaphors virtually does not depend at all on the position in the document (bin number). This can be explained by taking into account the overall structure of the NYT articles: before the TEXT section, they also contain PREAMBLE, where all the most important entities are introduced.

Consequently, when these entities are mentioned in the (beginning of the) text body, they are already anaphoric.

The second hypothesis is confirmed: the training data show a strong negative correlation ($R = -0.93$, $p < 0.01$) between the bin number and the percentage of +antecedents.

To validate the third hypothesis, we have computed the `paragraph rank` of our markables — the distance (measured in markables) to the beginning of the corresponding paragraph. Again, we normalize the counts by the paragraph's length and discretize them into 10 bins. Figure 6.2 (a) shows that `paragraph rank` affects anaphoricity only very mildly.

As predicted by our third hypothesis, the probability for a markable to be an antecedent decreases toward the end of a paragraph (Figure 6.2 (b), $R = -0.92$, $p < 0.01$). Most discourse theories investigate anaphoric accessibility for different markables within an utterance — a sentence or a clause. We see here that in bigger text units the distribution of antecedents also exhibits some regularity.

The same estimations can be done not on the paragraph, but on the sentence level. Corresponding results are shown on Figures 6.3 (for `sentence number`) and 6.4 (for `sentence rank`). The `paragraph number` and `sentence number` variables measure essentially the same parameter — the position of the markable in the whole text. It is therefore not surprising that the results are almost identical. The `sentence rank` variable shows a strong negative correlation with the percentage of both anaphors ($R = -0.96$, $p < 0.01$) and antecedents ($R = -0.98$, $p < 0.01$).

We have conducted the same experiments for the validation data. These articles, however, follow a very different discourse scheme — they are much longer (49, 69, and 30 sentences in the main text body, compared to 22 sentences on average for the training data) and switch between several topics. Unlike in the training data, we have not observed any significant correlation here. At the present stage of our research we do not have enough data to empirically analyze the structure of long multi-topic documents.

6.3 Discourse Levels: Embedded Sub-discourses

Linguistic theory considers a text to be "a unit of situational-semantic organization" (Halliday and Hasan, 1976): it is generally assumed that a single author produces a text on a single occasion within some period of time. This "continuous" view has several linguistic implications, such as, for example, the same interpretation for deictic expressions and situational anaphora ("this", "here", "today") throughout the document.

Real-world lengthy texts are often more complex and allow for such analysis only partially. Consider the following play, "The Mathematician and Andrey

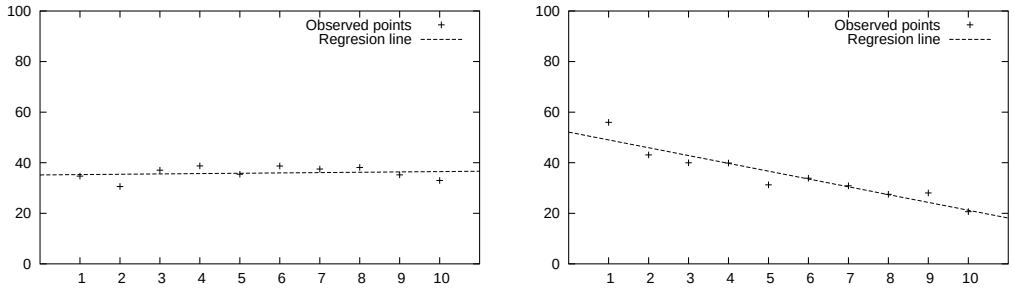


Figure 6.1: Percentage of anaphoric expressions (a) and antecedents (b) for different parts (normalized discretized paragraph number) of the main text body in the training data (30 "dry-run" MUC-7 texts).

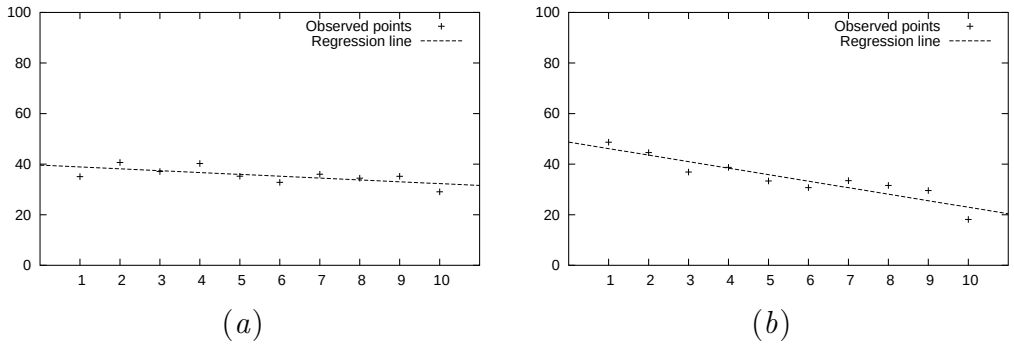
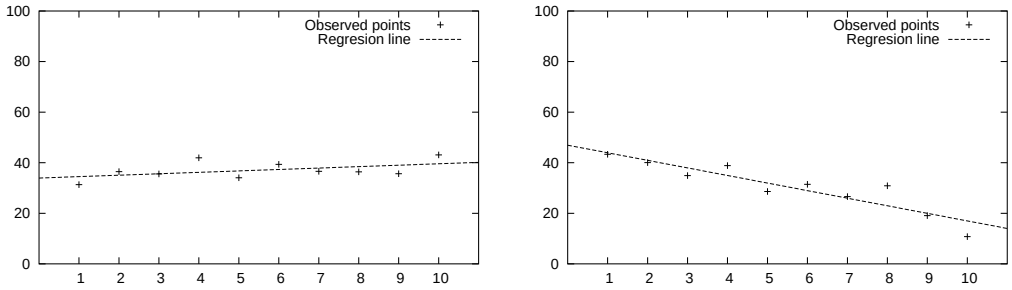


Figure 6.2: Percentage of anaphoric expressions (a) and antecedents (b) for different parts (normalized discretized distance to the beginning) of a paragraph in the training data (30 "dry-run" MUC-7 texts).

Semenovich" by Daniel Charms[4]:

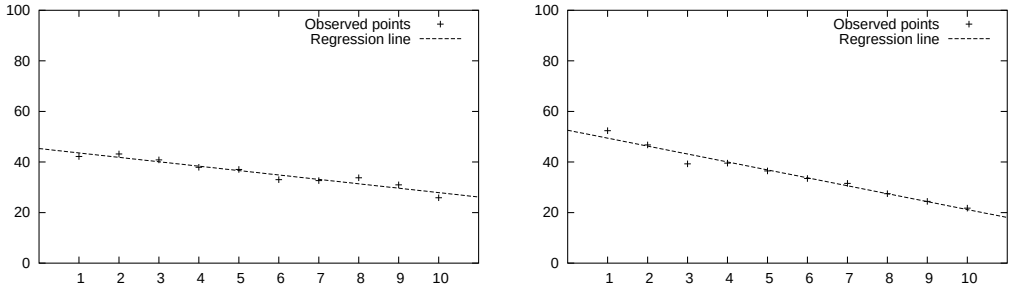
(70) *The Mathematician (pulling a sphere out of his head):*
I pulled a sphere out of my head.
Andrey Semenovich:
Put it back in.
M.: No, I won't.
A.S.: So, don't.
M.: So I won't.
A.S.: Whatever.
M.: So I won.

[4]Some sentences are repeated several times in the original version.



(*a*) (*b*)

Figure 6.3: Percentage of anaphoric expressions (a) and antecedents (b) for different parts (normalized discretized sentence number) of the main text body in the training data (30 "dry-run" MUC-7 texts).



(*a*) (*b*)

Figure 6.4: Percentage of anaphoric expressions (a) and antecedents (b) for different parts (normalized discretized distance to the beginning) of a sentence in the training data (30 "dry-run" MUC-7 texts).

> *A.S.:* You won, so relax.
> *M.:* No, I won't relax.
> *A.S.:* Although you are mathematician, to be honest, you are not very smart.
> *M.:* No, I am very smart and know a lot!
> *A.S.:* You know a lot, only it is not worth anything.
> *M.:* No, it's worth a lot.
> *A.S.:* I'm tired with this silliness.
> *M.:* No, you're not.
> *Andrey Semenovich waves his hand with sadness and leaves. The Mathematician waits for a minute and then goes after Andrey Semenovich.*

This play can be analyzed as a discourse produced by Daniel Charms (con-

tinuous view) or as a discourse produced by the Mathematician and Andrey Semenovich (discontinuous view). The former analysis regards the text as a whole and is helpful, for example, in investigating Charms' writing in general. The latter analysis splits the text into sub-discourses and is necessary to understand its content, for example, to resolve such anaphors as "you" or "I".

Newswire articles generally allow for the continuous analysis. However, even these short texts may contain sub-discourses — small dialogs or quoted opinions of other people.

Embedded discourses are problematic for coreference resolution. We have already seen in Section 4.9 that entities in quoted mini-texts often violate the agreement constraints — we repeat the relevant examples (50) and (51) below.

(71) "This doesn't surprise [me]$_1$ at all," said [Trish Neusch]$_{2,ante=1}$...

(72) [Provident]$_1$ Vice President Thomas White said , "[We]$_{2,ante=1}$ have said it is [our]$_{3,ante=1}$ strategy to grow [Provident]$_{4,ante=1}$".

Many other situational anaphors ("here", "tomorrow") can have different referents in the main text body and embedded parts. In addition, the author of a sub-discourse and the author of the whole document often make different assumptions on the information already presented to the reader/hearer and, thus, organize their messages in different ways. Consider the following example:

(73) About 25 percent of the airport traffic enters [a flight pattern]$_1$ over the plutonium storage bunkers.

The speaker assumes that "a flight pattern" is a new entity and uses the indefinite article "a". If (73) was a document on its own, "a flight pattern" would be the first NP in a coreference chain. In reality, this is an opinion quoted in a longer document and "a flight pattern" is already known to the reader:

(74) That 60-day accounting estimated that 25 planes a day entered [flight patterns]$_1$ over the zone where plutonium is stored in concrete bunkers covered with earth and grass ... About 25 percent of the airport traffic enters [a flight pattern]$_{2,ante=1}$ over the plutonium storage bunkers, McNulty said.

As a result, we see an anaphoric NP (not part of an apposition or copula construction) with an indefinite article, which is rather unusual for a homogeneous text.

To resolve such kinds of anaphors, we need two procedures: first, we have

to identify the boundaries of embedded sub-discourses, and, second, we need a mechanism to merge the discourse models of the main and embedded components. Both are known problems that have received much attention of the linguistic community (see below) and are still far from being solved.

State-of-the-art research on identifying sub-discourses is concentrated mainly around turn-taking in dialog processing (Ajmera et al., 2004) and opinion mining (Wiebe et al., 2003). Segmenting a spoken dialog into units is a phonetic problem outside the scope of this thesis. Written dialogs normally exhibit some structure (recall example (70) above), although it is still a non-trivial task to identify the speaker for each utterance, if this information is not marked explicitly. Opinion mining studies aim at identifying the Information Retrieval-motivated parameters of an opinion: its perspective (source) and attitude (polarity). Existing approaches normally operate at the sentence (Wiebe et al., 2003) or even text (Pang et al., 2002) level, without identifying exact boundaries of sub-discourses.

We have implemented a naive algorithm for the sub-discourse identification: first, we use a regular expression matcher to identify explicitly quoted opinions. In addition, we mine implicitly quoted opinions by searching for complex sentences with the verb *say* in their main clause. All the markables within a sub-discourse are marked +`embedded`.

(75) [The company will wait to receive the license before it builds], said David Margolese, chairman and chief executive.

This two-step procedure still misses various cases of sub-discourses,[5] for example:

(76) [But military training planes make up to 30 passes per flight over the plant], according to the Defense Nuclear Facilities Safety Board.

We hope that, with the increasing interest in opinion mining, representative annotated corpora will be soon available for researchers to empirically investigate this problem.

Combining the models of the main and embedded discourses is another non-trivial task. Most system tested on the MUC data do not attempt to solve

[5]In this study we are interested in sub-discourses that affect coreference. In many cases, as in example (76), an opinion is fully incorporated in the main text body and shares its discourse model. For example, personal pronouns "I" and "We" cannot be used in such sub-discourses:

"We will wait to receive the license before it builds", said David Margolese.

We will wait to receive the license before it builds, said David Margolese.

*We will wait to receive the license before it builds, according to David Margolese.

Such sub-discourses are not relevant for our approach.

	Anaphors				Antecedents			
Discourse level	+anaphor		−anaphor		+antecedent		−antecedent	
+embedded	250	41%	354	59%	224	37%	380	63%
−embedded	1453	33%	2971	67%	1479	33%	2945	67%

Table 6.2: Distribution of anaphors vs. non-anaphors and antecedents vs. non-antecedents for the main and embedded subdiscources in the training data (30 "dry-run" MUC-7 texts).

it and treat all the levels in a uniform way. An exception is the algorithm presented by Ng and Cardie (2002c), having a feature for explicitly quoted opinions. Most pronoun resolution algorithms try to avoid this problem as well: for example, many systems resolve only third person pronouns and do not cover the speaker-dependent cases of "You" and "I".

Several studies (Eckert and Strube, 2001; Tetreault and Allen, 2004) address the problem of pronoun resolution in (spoken) dialogue. However, these approaches are mainly concerned with task-specific difficulties, such as non-NP antecedents or "vague anaphora" (Eckert and Strube, 2001).

In the present study, we do not propose a specific algorithm for anaphora resolution in multi-level discourses: with almost no data available, we cannot address this problem empirically. However, as a first attempt, we have two kinds of relevant features. First, we relax agreement constraints for embedded discourses (see Sections 4.9 and 5.2 for corresponding features). Second, we encode as a boolean feature the fact that a markable appears in an embedded sub-discourse. As Table 6.2 shows, markables in embedded sub-discourses tend to be more often anaphors and antecedents than those in the main text body (χ^2-test, $p < 0.01$). This is a genre-specific property of newswire articles — quoted opinions are often used to back up some claims of the main text's author and usually repeat already known entities.

6.4 Proximity

Although newswire texts are mainly devoted to one topic, they still talk about it from different angles, introducing new entities and abandoning material that has already been sufficiently represented. Therefore anaphors and antecedents are usually close to each other. The *proximity* factor is especially crucial for pronominal anaphora — most pronoun resolution algorithms restrict their search space to a 2-5 sentences window and impose distance-dependent penalties on candidates. In this section we investigate the influence of proximity on NP-coreference.

From a computational perspective, proximity information can be incorpo-

rated into an algorithm implicitly or explicitly. In the first case, we resolve anaphors by checking all the candidate antecedents from right to left, thus, rewarding markables close to the anaphor:

(77) News of the talks was reported in today's Wall Street Journal. Provident stock fell 1/8 to 30 7/8 in early trading. Textron shares rose 3/4 to 88 7/8.
<p>
Officials at [Paul Revere]$_1$ weren't available for comment. [Its]$_{2,ante=1}$ shares rose 3/4 to 25.

In this example we have several competing candidates for the pronoun "Its": "News", "the talks",... By submitting them to a resolution algorithm one by one, starting from the very end and going to the beginning, we find the correct one, "Paul Revere", without ever seeing other markables.

In the second case, we introduce some measure of proximity and encode it as a feature for a learning-based algorithm or as (a part of) a rule for a hand-crafted rule-based system.

The implicit solution relies on the first-link clustering: as soon as we have found a suitable antecedent, we link our anaphor to it and proceed to the next markable. Several recent studies (Ng and Cardie, 2002c; McCallum and Wellner, 2003; Luo et al., 2004) show that this clustering strategy is too local and that much better coreference resolution systems can be built by shifting to a more global clustering strategy.

The explicit solution can lead to problems in a machine learning approach for sampling strategies that do not rely on hard constraints to restrict the search space ("window"). For example, in the commonly used set-up (Soon et al., 2001), the training set is constructed by pairing each anaphor with its closest antecedent and all the markables in between. In this case, −antecedents are by definition closer to the anaphor than +antecedents.

We have conducted an experiment to check if proximity interacts with coreference in our data. If we assume that the proximity factor plays no role in interpretation of anaphors, we may say that the probability p of two markables being coreferent is independent on the distance between them. In this case we can model coreference resolution as a Bernoulli process with the success probability p: for each anaphor, we check the candidate antecedents from right to left and resolve it to each candidate with the probability p. The distance from the anaphor to its closest antecedent corresponds in this model to the number N of trials till the first success — geometric distribution with parameter p:

$$P_N(n) = (1-p)^{n-1}p,\ n = 1, 2\ldots,$$

$$E(N) = 1/p.$$

We use the corpus data to estimate p for different metrics of proximity: `markable distance`, `sentence distance`, and `paragraph distance`. They are measured as follows: markables are extracted by our preprocessing module (Section 2.5), sentence boundaries are identified by a Maximum-Entropy tool (Reynar and Ratnaparkhi, 1997), and paragraphs are annotated in the original MUC data. In our example (77), the distance between "Its" and "Paul Revere" is 2 in markables, 1 in sentences, and 0 in paragraphs.

Figure 6.5 shows the number of anaphors with their closest antecedents at `markable distance` n in the training data. The corresponding estimated geometric distributions are shown with dashed lines. The χ^2-test suggests ($p < 0.01$, all the entities with $n > 20$ are merged into one class to fulfill the assumptions of the χ^2-test) that these samples do not come from the geometric distribution. We see that corpus counts go down much steeper in the beginning: at distances of $n < 5$, the observed frequencies are higher than the expected ones. This shows that the proximity factor interacts with coreference: the closer markables are much more likely to be true antecedents for a given anaphor.

Figures 6.6 and 6.7 show the same distribution for distances measured in sentences and paragraphs. Cases of intrasentential and intra-paragraph coreference are removed to facilitate the computation of the estimated distribution[6]. Again, the χ^2-test clearly shows ($p < 0.01$) that the observed distributions are not geometric. This gives an empirical support to the hypothesis that proximity interacts with coreference.

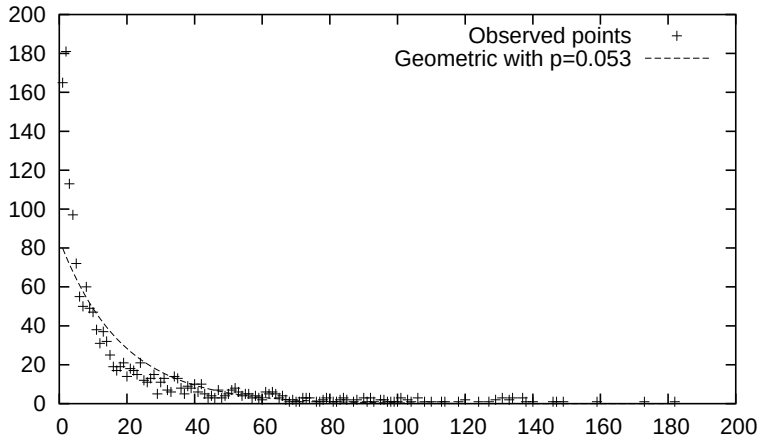


Figure 6.5: Number of anaphors in the training data (30 "dry-run" MUC-7 texts) for different markable distances to their closest antecedents.

[6]The memorylessness property of the Bernoulli process makes it a valid procedure: if we start it at the point $x + y$ instead of x, the number of trials till the first success is again distributed geometrically.

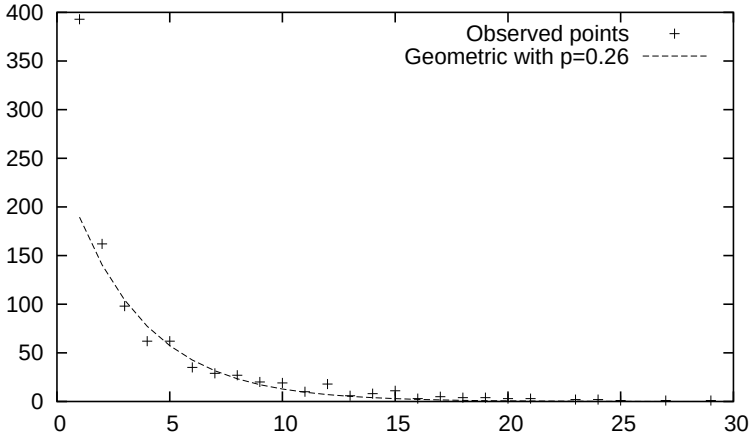


Figure 6.6: Number of anaphors in the training data (30 "dry-run" MUC-7 texts) for different sentence distances to their closest antecedents.

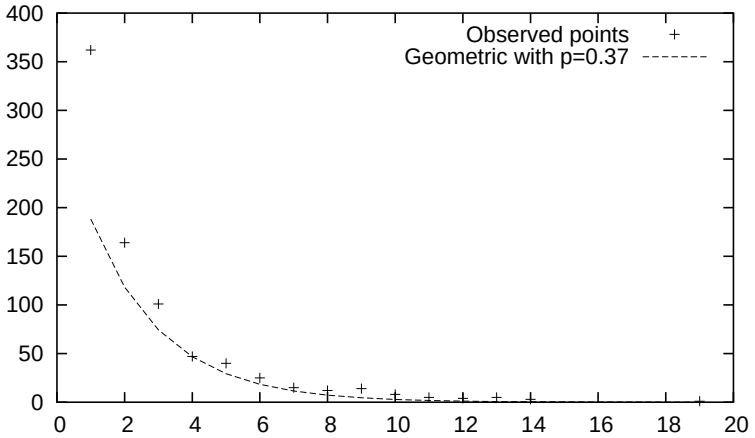


Figure 6.7: Number of anaphors in the training data (30 "dry-run" MUC-7 texts) for different paragraph distances to their closest antecedents.

Distance	+coreferent		−coreferent	
+same_sentence	609	4%	16440	96%
−same_sentence	5377	2%	311479	98%
+same_paragraph	970	3%	27399	97%
−same_paragraph	5016	2%	300520	98%

Table 6.3: Distribution of coreference links for intra- and intersentential and intra- and inter-paragraph pairs in the training data (30 "dry-run" MUC-7 texts).

Cases of intrasentential and intra-paragraph coreference require special treatment. Table 6.3 shows that markables from the same sentence or paragraph tend to be more often coreferent than distant ones (χ^2-test, $p < 0.01$).

6.5 Centering-based Discourse Properties

Centering (Joshi and Weinstein, 1981; Grosz et al., 1983; Grosz et al., 1995) is a well-established linguistic theory of discourse coherence and salience. Its main goal is to identify cross-lingual structural preferences that make some discourses easier to process.

Numerous studies in various fields have been based on or motivated by the ideas of centering. Psycholinguists (Hudson et al., 1986; Gordon et al., 1993) have conducted behavioral experiments verifying the main claims of centering. Computational linguists have proposed centering-based algorithms for anaphora resolution (Brennan et al., 1987; Tetreault, 2001) and generation (Henschel et al., 2000). Theoretical linguists have investigated and applied centering principles to different languages (Kameyama, 1985; Walker et al., 1994).

Extensive corpus-based studies (Tetreault, 2001; Poesio et al., 2004), however, suggest that centering ideas should not be taken per se, but in combination with many other factors. Thus, Tetreault (2001) shows that pure centering-based pronoun resolution algorithms cannot outperform the naive syntax-based approach of Hobbs (1978). However, augmented with a procedure for intrasentential coreference resolution, the centering approach of Tetreault (2001) achieves promising results.

The empirical study of Poesio et al. (2004) shows that the main claims of centering concerning coreference (in particular, pronominalization) are robust, but very weak: the suggested rules and constraints are not applicable in most cases. For example, the classification of CF transitions (see below) is only applicable when two adjacent sentences share at least one entity, which seldom happens in the corpus of Poesio et al. (2004). Consequently, they characterize centering principles as important but not the only factors of discourse salience

and coherence.

In our system, we represent the key concept of the centering theory, the backward-looking center (CB), as a feature[7], thus, combining it with other kinds of information in a machine-learning set-up.

A discourse consists of a sequence of *utterances* $U_1, \ldots, U_n$. Each utterance U_i is associated with a ranked *list of forward looking centers*, $CF(U_i)$ — discourse entities that are directly realized in the utterance. The ranking strategy is an open research issue addressed differently in each approach (most studies advocate syntax-based ranking schemes). The first entity of the $CF(U_i)$ list is called the *preferred center*, $CP(U_i)$. The highest-ranked element of $CF(U_i)$ that is also realized in U_{i-1} is called the *backward-looking center*, $CB(U_i)$.

The update of the CF list when processing a new sentence, known as its *transition*, determines how much cognitive effort a hearer needs to process it. Most papers identify two factors for the typology of transitions: (1) whether or not $CB(U_i)$ is the same entity as $CB(U_{i-1})$ and (2) whether or not $CB(U_i)$ is the same entity as $CP(U_i)$. Table 6.4 introduces the main transition types, as proposed in (Brennan et al., 1987)[8].

	$CB(U_i) = CB(U_{i-1})$	$CB(U_i) \neq CB(U_{i-1})$
$CB(U_i) = CP(U_i)$	CONTINUE	SMOOTH SHIFT
$CB(U_i) \neq CP(U_i)$	RETAIN	ROUGH SHIFT

Table 6.4: CF transitions proposed by the Centering theory.

The centering theory proposes several principles determining and relating various properties of CF, CP, and CB. Poesio et al. (2004) argue that not all of them have the same status, distinguishing between definitions and claims. The following three main principles are claims that can be verified empirically:

Constraint 1: All utterances except for the first one have exactly one CB (strong version, the weak formulation of the same constrain requires at most one CB).

Rule 1: The CB is pronominalized always (strong) or if any other markable in the utterance is pronominalized (weak).

Rule 2: Continuations are preferred over retains over shifts.

[7]As shown in (Poesio et al., 2004), centering should be better viewed as a framework or a parametric theory with various concepts ("utterance", "ranking",...) to be specified. In the present study we mainly follow the mainstream specification, simplifying it to obtain reliable results with our shallow preprocessing modules: we consider sentences and not clauses to be utterances.

[8]The SMOOTH SHIFT transition is called SHIFT-1 in (Brennan et al., 1987)

Different papers on centering propose variants of these claims. An empirical evaluation for both the strong and the weak versions is presented in (Poesio et al., 2004).

The claims suggest that CBs are important entities ensuring local coherence. By definition, CBs are always anaphors. Rule 2 states that CBs tend to be antecedents for some anaphors in the next sentence. Such knowledge is essential for coreference resolution and therefore we have conducted an evaluation experiment to investigate the properties of CBs in our data.

To compute CBs in the training corpus, we rely on the MUC-7 annotation for determining which elements of CF are realized (i.e. have antecedents) in the previous sentence. For the test data, we rely on the system's output for already resolved sentences: as the documents are processed from left to right, we can always determine if a candidate antecedent is a CB or not.[9] In both cases, grammatical roles (Section 4.8) are used to rank the CF list.

First, only 45.7% of sentences in our training corpus have a CB. A similar percentage was observed by Poesio et al. (2004). This is discouraging: for more than half of the sentences, centering principles are not applicable at all. Second, most sentences are very long and therefore contain many intrasentential anaphors — full NPs (78) or pronouns (79):

(78) Dorn notes that the get-tough policy of the Franklin D. Roosevelt administration toward $[\text{Japan}]_1$ — designed to persuade $[\text{Japan}]_{2,ante=1}$ to rein in $[\text{its}]_{3,ante=2}$ military expansionism in China and Southeast Asia — included an embargo against oil exports to $[\text{Japan}]_{4,ante=3}$ and the freezing of Japanese assets in the summer of 1941.

(79) With one hurricane recently departed and another one stirring the Atlantic anew, conditions were so bad on Thursday morning, $[\text{Francis}]_1$ said, that a wave knocked $[\text{him}]_{2,ante=1}$ into the water as it swept the Navy speedboat carrying $[\text{him}]_{3,ante=2}$ and several other investigators to the Grapple.

This makes centering information less relevant and increases the role of syntactic knowledge. However, as Table 6.5 shows, `cb` is a very good predictor for antecedents (χ^2-test, $p < 0.01$): 70% of CBs in the training data (and even 81% in the validation data) are +antecedents.

In sum, the results are ambivalent: on the one hand , the `cb` feature is a reliable indicator for antecedents, on the other hand, it fires not very often. In the next session, we present several combinations of `cb` with positional and agreement features.

[9] The same strategy is used for other "recursive features", see Section 6.7.

CB	+antecedent		−antecedent	
+CB	206	70%	88	30%
−CB	1278	31%	2807	69%

Table 6.5: Distribution of antecedents vs. non-antecedents for ±CB in the training data (30 "dry-run" MUC-7 texts).

6.6 Salient Candidates

As we have seen in the previous sections, several factors may affect the probability of a markable being an antecedent. They include proximity (see Section 6.4), position in the sentence or paragraph (Section 6.2), grammatical functions (Section 4.8), and CB (Section 6.5).

We use these properties to compute a set of "likely antecedents" for a given anaphor. They can be further filtered by the agreement constraints – `syntactic_agreement_quoted` (see Section 4.9).

A family of boolean features describes the pool of likely antecedents. Each feature $f(anaphor, antecedent)$ is represented as a triple $\{proximity, salience, agreement\}$. We use the following functions to measure proximity: ±`same` (the anaphor and the antecedent are in the same sentence), ±`prev` (the anaphor and the antecedent are in adjacent sentences), ±`closest` (the antecedent is the closest markable to the anaphor satisfying the specified *agreement* and *salience* conditions). Salience is encoded by ±`subject`, ±`ssubject` (see `sentence subject` in Section 4.8), ±`cb`, ±`closest`, ±`sfirst` (the first markable in a sentence), and ±`pfirst` (the first markable in a paragraph). Finally, agreement can be triggered on (encoded by `agree` in the feature name) or off (not encoded in feature names).

For example, `prev_cb(`M_i`,`M_j`)` is a boolean function indicating that M_j (a candidate antecedent) is a CB of some sentence, and M_i (an anaphor) is a markable from the next sentence; `closest_first_agree(`M_i`,`M_j`)` indicates that M_j is (1) a first markable in some sentence, (2) has compatible agreement values with M_i, and is the closest markable to M_i satisfying the conditions (1) and (2); `closest_closest_agree(`M_i`,`M_j`)` indicates that M_j is the closest to M_i markable with the matching agreement values.

In the example below, `prev_cb` is true for the pair ("these experts", "Hellenikon International Airport"), `closest_first_agree` is true for the pair ("these experts", "Airline and airport staffs"), and `closest_closest_agree` is true for the pair ("these experts", "security precautions"):

(80) In March, the Federal Aviation Administration issued a travel warning for [the airport]$_1$, but lifted it in May after a new training program for airport police began. [Airline and airport staffs based in Athens and New

> York]$_2$, along with aviation security experts, say that [Hellenikon International Airport]$_{3,ante=1}$ in Athens has more [security precautions]$_4$ in effect than Kennedy International Airport. In fact, [these experts]$_{5,ante=2}$ say, European airports have gone much further toward combating the threat of explosives on airplanes than airports in the United States.

Table 6.6 (left part) shows the distribution of coreference links for the proposed functions. In total, we have 333905 links (5986 +coreferent and 327919 −coreferent) in the training and 154291 links (2106 and 152185) in the validation data. All the functions except from `closest_prev`, the closest markable from the preceding sentence, significantly ($p < 0.01$) affect the distribution and therefore are potentially relevant for coreference resolution. Triggering the agreement on results in better predictions: the number of true positives remains essentially the same, but the number of false positives decreases drastically.

We have already mentioned that discourse properties are more important for pronoun resolution. Therefore we have investigated the interaction between our functions and coreference properties of pronominal anaphors. We took the same data and discarded all the pairs with the non-pronominal first markable (M_i). The results are presented in the right part of Table 6.6. Again, all the functions except from `closest_prev` significantly ($p < 0.01$) affect the distribution in both the training and validation data.

Altogether, our functions cover 1149 of 5986 (19%) +coreferent links. More importantly, for 43.3% of all the anaphors in the training data (49.9% for the validation set), their closest antecedents belong to the pool of "likely antecedents" identified by the functions. For pronominal anaphors the numbers are even higher: 24% of the links are covered by at least one function and 80.9% pronouns (73.2% for the validation data) have their closest antecedents in the corresponding pool. This means that our set of "likely antecedents" can effectively be used to guide the search, especially for pronouns.

6.7 Coreferential Status of Candidate Antecedents

Theoretical studies suggest a tight interaction between discourse structure and anaphoric accessibility, introducing the notion of *salience*. In the previous sections we have always considered salience to be a property of a *markable*, saying, for example, that "markables in the beginning of a paragraph are more salient (tend to be antecedents)". This, on the one hand, allows us to compute straightforwardly basic salience-related factors. On the other hand, this view is too local and deviates from most theoretical definitions, that regard salience as a property of *discourse entities*, e.g. coreference chains.

In this section we investigate the possibilities to go back to the more global

Salient Candidate	All pairs +coref		All pairs −coref		Pronominal anaphora +coref		Pronominal anaphora −coref	
all links	5986	2%	327919	98%	2104	6%	31255	94%
+CB_same	104	8%	1186	92%	61	42%	83	58%
+CB_prev	147	8%	1743	92%	60	31%	135	69%
+CB_closest_agree	256	8%	2950	92%	118	36%	214	64%
+CB_same_agree	99	11%	769	89%	57	54%	48	46%
+CB_prev_agree	145	12%	1052	88%	59	44%	75	56%
+subject_same	386	6%	5976	94%	213	34%	405	66%
+subject_prev	405	5%	8184	95%	187	21%	689	79%
+subject_closest_agree	332	9%	3546	91%	186	45%	225	55%
+subject_same_agree	356	9%	3407	91%	199	48%	213	52%
+subject_prev_agree	376	8%	4608	92%	171	31%	379	69%
+ssubject_same	58	6%	950	94%	36	24%	111	76%
+ssubject_prev	33	3%	931	97%	16	17%	80	83%
+ssubject_closest_agree	134	4%	3342	96%	57	16%	303	84%
+ssubject_same_agree	54	9%	539	91%	34	35%	64	65%
+ssubject_prev_agree	32	6%	476	94%	16	26%	46	74%
+sfirst_same	192	5%	3518	95%	95	30%	220	70%
+sfirst_prev	186	5%	3791	95%	84	22%	301	78%
+sfirst_closest_agree	292	7%	3826	93%	141	35%	264	65%
+sfirst_same_agree	182	8%	2075	92%	91	43%	122	57%
+sfirst_prev_agree	170	7%	2132	93%	76	33%	157	67%
+pfirst_same	120	5%	2284	95%	57	31%	129	69%
+pfirst_prev	126	6%	2128	94%	61	25%	182	75%
+pfirst_closest_agree	262	7%	3675	93%	121	30%	277	70%
+pfirst_same_agree	111	7%	1384	93%	53	41%	75	59%
+pfirst_prev_agree	114	9%	1213	91%	55	37%	93	63%
+closest_same	148	4%	3585	96%	29	9%	287	91%
+closest_prev	84	2%	3925	98%	30	8%	356	92%
+closest_closest_agree	166	5%	3019	95%	99	24%	314	76%
+closest_same_agree	129	5%	2294	95%	77	27%	212	73%
+closest_prev_agree	81	3%	2356	97%	28	12%	214	88%

Table 6.6: Distribution of ±coreferent links for basic salience functions in the training data(30 "dry-run" MUC-7 texts): all the pairs (left) and only pairs with pronominal anaphors (right) considered.

view and extract and encode the information about entities. In other words, we are interested in coreference properties of candidate antecedents: the size of their chains and the (markable-based) salience of their antecedents.

Most symbolic approaches to (pronominal) anaphora resolution pay close attention to coreference properties of candidate antecedents. For example, the Centering theory ((Grosz et al., 1995), see also Section 6.5 above for a very brief summary) makes a claim (Rule 2) about transitions, tracking the entities in several subsequent utterances. Correspondingly, centering-based pronoun resolution algorithms (Brennan et al., 1987; Tetreault, 2001) check if a candidate antecedent and *its antecedent* are both CBs and/or CPs.

Machine learning approaches (we mention two notable exceptions below), on the contrary, do not encode any coreferential properties of antecedents. Some algorithms try to resolve all the entities in the same time, recasting coreference resolution as a task of building clusters (Cardie and Wagstaff, 1999) or sequences (McCallum and Wellner, 2003) of entities. In such settings, coreferential properties of candidates are not known. Most systems, however, process texts from left to right, and, thus, could directly incorporate such knowledge, but still ignore it. This can potentially be dangerous:

(81) [McDonald's Corp.]$_1$ is shopping for customers inside some of the nation's biggest retailers, including Wal-Mart Stores Inc. and Home Depot Inc. And why not, since 75 percent of [McDonald's]$_{2,ante=1}$ diners decide to eat at [its]$_{3,ante=2}$ restaurants less than five minutes in advance?

The first markable, "McDonald's Corp.", is very salient: it is a subject, the first NP of a sentence and a paragraph (and its sentence does not have a CB at all). The second markable in the chain, "McDonald's", on the contrary, is not salient at all. If we applied a right-to-left algorithm to resolve "its", we probably wouldn't consider "McDonald's" as a good candidate. Having a reliable system, we still can hope to build a correct chain by resolving "its" to "McDonald's Corp." directly. However, to do so, we have to reject all the intervening markables, which might be problematic: for example, "75 percent" is another salient entity that might misguide the system.

The problem can be alleviated if we could incorporate some information to tell the system, that, although "McDonald's" is not a salient markable by itself, it represents a very salient entity.

In this chapter, we investigate the interaction between various coreferential properties of a markable and its probability to be an antecedent for some anaphor. Our experiments are motivated by the following two studies. Ge et al. (1998) rely on the repetition count to encode global salience. Yang et al. (2004) present a system for pronoun resolution augmented with several features for coreferential properties of candidates. They show that such features boost the system's performance dramatically in an oracle setting and

Information status	+antecedent		−antecedent	
discourse old	1020	64%	569	36%
discourse new	464	17%	2326	83%

Table 6.7: Distribution of antecedents vs. non-antecedent for discourse old and discourse new markables in the training data (30 "dry-run" MUC-7 texts).

still increase it in a more realistic scenario.

In our study, we encode coreferential properties of candidate antecedents in several features: information status of the antecedent, repetition count, and candidate's antecedent's (ante_ante) parameters. To compute these values we have to know the correct chains. For the training data, we take the chains from the MUC annotation. During testing, we process the text from left to right, resolving all the markables on the way. Consequently, when encountering an anaphor, we already have constructed the chains for all the possible candidates. The important difference between the two procedures is that our chains for the training data are perfect, whereas the ones for the testing data are constructed automatically and therefore error-prone.

Information status of the antecedent. Several studies suggest that discourse old entities are more likely to be antecedents. This is an explicit claim of Strube and Hahn (1999) and Strube (1998) and also a corollary of the Rule 2 of the centering theory.

We encode the information status (old vs. new) in a boolean feature[10]: `long chain` is set to 1 if the candidate antecedent is a discourse old entity (that is, it is itself an anaphor): in such a situation, if we link our anaphor to the candidate, we would obtain a chain of at least 3 entities.

Table 6.7 shows that information status is a very reliable predictor of antecedents — 64% of discourse old entities (75% for the validation data) are mentioned once again. Compared to the other salience-increasing factors investigated in this study, information status is much more robust — our data contain five times more discourse old entities than, for example, CBs (see Table 6.5).

In Section 6.2 we have formulated several hypotheses on the structure of a short newswire article. They suggest that short coreference chains should more

[10]Strube (1998) and Yang et al. (2004) propose special mechanisms to account for information status of the parts of appositive constructions. They argue that appositions relate entities to the hearer's knowledge and, thus, make them discourse old. In our approach, we do not need any extra procedure here, as both parts of appositions are treated as markables: by the time the whole apposition is processed, our algorithm has built a chain out of its parts and the corresponding entity becomes discourse old automatically.

often appear within a single paragraph than to span over several ones. In our corpus, in short chains 48% of the anaphors (248/521) have intra-paragraph antecedent, compared to 41% in long chains (426/1048). This difference is significant ($p < 0.01$). Moreover, the average `paragraph distance` between two markables in a long chain is 2.39 and in a short chain 3.5. This means that long chains roughly correspond to main topics repeated again and again throughout the text. Short chains, on the contrary, generally represent local topics or "outlier" cases of long-distance coreference.

Repetition. Local discourse factors may increase the salience level of some candidates — CBs, subjects, first NPs in a sentence or paragraph. However, every short text is devoted to a few central topics that are always salient. Global salience can be estimated by the number of times an entity has been mentioned throughout the document, that is, its chain size.

Earlier papers on pronominal Anaphora Resolution have taken antecedents' chains into account implicitly (Lapin and Leass, 1994; Kennedy and Boguraev, 1996) or explicitly (Ge et al., 1998). Later studies, however, ignored this knowledge, as it is very difficult to obtain reliably without having a full-scale Coreference Resolution algorithm.

Figure 6.8 shows for the training data the percentage of +antecedents among entities mentioned n times in a document. We see that discourse new entities ($n = 0$) are very seldom antecedents and the more often an entity has already been mentioned, the higher the probability ($R = 0.74$, $p < 0.01$) that it will be mentioned once again.

The same tendency holds within a paragraph. Figure 6.9 shows the percentage of antecedents among entities mentioned n times within a single paragraph. Again, we see a correlation here ($R = 0.86$, $p < 0.01$).

Candidate's antecedent Coreference links can be represented not only as $(anaphor, antecedent)$ pairs, but also as $(anaphor, antecedent, ante\_ante)$ triples, where the *ante_ante* element is the closest correct antecedent of *antecedent*. Recall that for training and testing we have (different) procedures to compute ante_ante.

Several boolean features encode salience parameters of candidates' antecedents (ante_ante). These features are motivated by the work of Yang et al. (2004) on pronominal anaphora.

Similar to Section 6.6, we check if ante_ante is (a) a CB, (b) a subject, (c) a subject of a main clause , (d) a first NP in a sentence, or (e) a first NP in a paragraph. Table 6.10 shows the distribution of ±antecedents depending on the salience properties of ante_ante.[11] All the factors affect the distribution

[11]For example, of 218 markables having CBs as their closest antecedents, 166 are +antecedents and 52 are −antecedents.

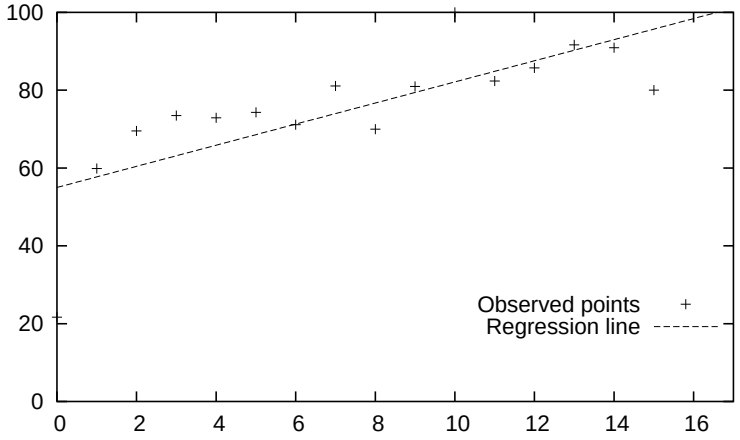


Figure 6.8: Percentage of antecedents for entities mentioned n times in a document, training data (30 "dry-run" MUC-7 texts).

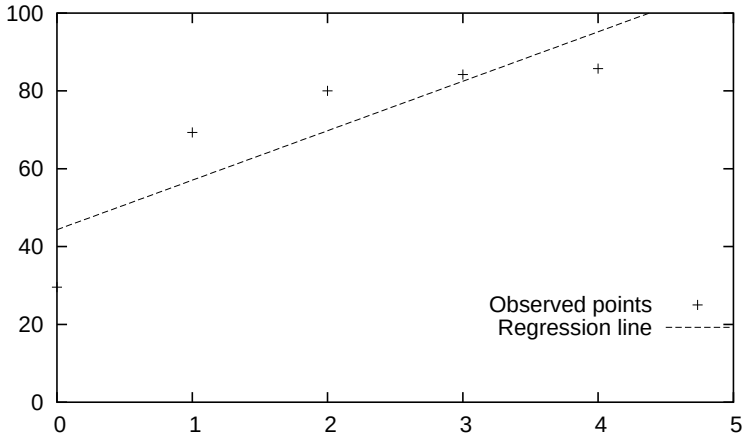


Figure 6.9: Percentage of antecedents for entities mentioned n times in a paragraph, training data (30 "dry-run" MUC-7 texts).

ante_ante marktype	+antecedent		−antecedent	
DEFNP	208	64%	118	36%
DETNP	36	60%	24	40%
PRON	209	77%	64	23%
NE	380	68%	179	32%
OTHER	187	50%	184	50%

Table 6.8: Distribution of antecedents vs. non-antecedent for `markable type` of candidate's antecedent (ante_ante) in the training data (30 "dry-run" MUC-7 texts).

ante_ante neclass	+antecedent		−antecedent	
DAT	20	50%	20	50%
LOC	63	54%	53	46%
ORG	125	72%	49	28%
PER	108	75%	36	25%
TIM	0	0%	4	100%
O	680	63%	402	37%

Table 6.9: Distribution of antecedents vs. non-antecedent for `named entity class` of candidate's antecedent (ante_ante) in the training data (30 "dry-run" MUC-7 texts).

significantly (χ^2-test, $p < 0.01$).

In Sections 4.2 and 5.2 we have investigated two surface properties of markables that may influence their probability of being +antecedent: `type of markable` and `neclass`. We also have corresponding features for ante_ante, interacting with the distribution of antecedents (Tables 6.8 and 6.9, χ^2-test, $p < 0.01$). We see that if an entity, expressed by a pronoun or a name of ORGANISATION/PERSON, is mentioned once again, it is very likely (around 75% percent for the training data) to be repeated for the third time. For comparison, the probability of an entity expressed by a pronoun to be mentioned for the second time is much lower — 64.5% (see Section 4.2)

This statistics shows that a coreference resolution system might benefit a lot from incorporating anaphoric properties of candidate antecedents. However, it must be taken into account that in this section we have used the MUC annotation to compute the values for our functions. To analyze the test data, we have to obtain this information recursively from our own algorithm, relying on already constructed chains at each processing step. It is not a priori clear whether these more noisy features will improve or deteriorate the system's overall performance. We will address this issue in the next section.

ante_ante	+antecedent		−antecedent	
+subject	478	68%	228	32%
−subject	1006	27%	2667	73%
+sent_subject	86	68%	41	32%
−sent_subject	1398	33%	2854	67%
+sfirst	236	69%	107	31%
−sfirst	1248	31%	2788	69%
+pfirst	147	71%	59	29%
−pfirst	1337	32%	2836	68%
+cb	166	76%	52	24%
−cb	1318	32%	2843	68%

Table 6.10: Distribution of antecedents vs. non-antecedent for different salience parameters of candidate's antecedent in the training data (30 "dry-run" MUC-7 texts).

6.8 Experiments

In the previous sections we have investigated various discourse properties of markables that can potentially affect coreference. We have encoded these factors as features for a machine learning approach. In this Section we evaluate the contribution of our salience-based features. The SVM^{light}learning package (Joachims, 1999) has been used in all our experiments.

Altogether we have 97 discourse and salience-based features listed in Table 6.11. As an illustration, we also show two feature vectors – for the pairs ("its"$_6$, "The Navy"$_4$) and ("The Navy"$_{20}$, "it"$_{10}$) in the following example:

(82) <TEXT>
<p>
[Washington]$_1$, [Feb. 22]$_2$ -LRB- [Bloomberg]$_3$ -RRB-
[The Navy]$_4$ ordered [its]$_{6,ante=4}$ [Northrop Grumman Corp.]$_7$ F-14s]$_5$ out of [the skies]$_8$ for [three days]$_9$ while [it]$_{10,ante=6}$ investigates [three recent F-14 crashes]$_{11}$.
<p>
[The action]$_{12}$ came after [an F-14A fighter]$_{13}$ crashed [today]$_{14,ante=2}$ in [the Persian Gulf]$_{15}$. [The two crew members]$_{16}$ ejected and were rescued and returned to [the aircraft carrier]$_{17}$ [USS Nimitz]$_{18}$ with [minor injuries]$_{19}$, [the Navy]$_{20,ante=10}$ said.
<p>
...

We have seen throughout this chapter that some features do not significantly

affect coreference. More specifically, many features describe the most salient *antecedent* and are not relevant for *anaphors*. Some features have more efficient variants: for example, our boolean features for the candidates pool (Section 6.6) show much better quality when agreement is triggered on. We call these less relevant properties "secondary" features. Altogether we have 28 secondary features shown in Table 6.11 in italic.

Most features can be extracted straightforwardly from the output of the preprocessing modules. However, the backward looking center (Section 6.5) and coreferential parameters of candidate antecedents (Section 6.7) can only be extracted using the system's output in a recursive way: we have to resolve the antecedent and all the preceding markables to be able to compute these values. Such features (22 in total) are shown in boldface in Table 6.11. The remaining 52 features, the "main" subset, are shown in plain font.

Recursive features can boost the performance of a good algorithm but completely damage a bad algorithm: if a candidate antecedent has been resolved incorrectly, the values of such features become erroneous starting a chain reaction. For example, if our algorithm has a too strong preference for intersentential candidates, "its" in 82 would be resolved to "Washington". Then, when resolving "it", the algorithm would rely on too high salience values for "it" (it would be a discourse old entity with a very salient ante_ante), probably merging "Washington", "its", and "it" into one chain. At the next step, "its" would become a very plausible candidate for "The action" and so on. All this would not happen if we had no recursive features: the initial mistake (resolving "its" to "Washington") would not influence further processing.

6.8.1 Experiment 7: Salience-based Pronoun Resolution

In the beginning of this chapter we have mentioned that pronominal anaphora is an ideal testbed for salience-based approaches to coreference. Both linguistic and psychological studies claim that context factors are crucial for pronominal anaphora interpretation. From a computational perspective, most matching, syntactic or semantic criteria are not applicable to pronouns: for example, differences in their surface form (*you* vs. *your* and so on) can be better accounted for directly than with name-matching techniques proposed above.

In this experiment we use our validation data, 3 annotated MUC-7 "training" documents to evaluate the performance of different salience-based feature sets for pronominal coreference resolution.[12]

[12]We have discarded non-pronominal anaphors from the annotation. This resulted in 135 anaphors for the validation data.

Table 6.11: Discourse and salience-based features and their values for the pairs ("its"$_6$, "The Navy"$_4$) and ("The Navy"$_{20}$, "it"$_{10}$) in example 82. Secondary features are shown in italic and recursive features — in boldface.

Feature	Range	$pair_1$ value	$pair_2$ value
Anaphor's parameters			
section_tag(M_i)	slug, date, nwords, preamble, text, trailer	text	text
paragraph_number_bin(M_i)	1...10	1	2
sentence_number_bin(M_i)	1...10	1	1
paragraph_rank_bin(M_i)	1...10	5	10
sentence_rank_bin(M_i)	1...10	2	10
embedded(M_i)	0,1	0	0
cb(M_i)	0,1	0	0
subject(M_i)	0,1	0	1
sentence_subject(M_i)	0,1	0	1
first_in_sentence(M_i)	0,1	0	0
first_in_paragraph(M_i)	0,1	0	0
Antecedent's parameters			
section_tag(M_j)	slug,date,nwords, preamble, text, trailer	text	text
paragraph_number_bin(M_j)	1...10	1	1
sentence_number_bin(M_j)	1...10	1	1
paragraph_rank_bin(M_j)	1...10	3	9
sentence_rank_bin(M_j)	1...10	1	9
embedded(M_j)	0,1	0	0
cb(M_j)	0,1	0	0
subject(M_j)	0,1	1	1
sentence_subject(M_j)	0,1	1	0
first_in_sentence(M_j)	0,1	1	0
first_in_paragraph(M_j)	0,1	0	0
ante_discourse_old(M_j)	0,1	0	1
chains_size(M_j)	continuous	0	2
chains_size_same_paragraph(M_j)	continuous	0	2
ante_ante_marktype(M_j)	−, defnp, ne, pro, other	−	pro
ante_ante_netag(M_j)	−, dat, loc, org, per, tim, none	−	none
ante_ante_sent_subject(M_j)	0,1	0	0
ante_ante_subject(M_j)	0,1	0	0
ante_ante_sfirst(M_j)	0,1	0	0
ante_ante_pfirst(M_j)	0,1	0	0

Table 6.11: (continued)

Feature	Range	$pair_1$ value	$pair_2$ value
ante_ante_cb(M_j)	0,1	0	0
Pair's parameters, all anaphors			
paragraph_distance(M_i,M_j)	continuous	0	1
sentence_distance(M_i,M_j)	continuous	0	2
markable_distance(M_i,M_j)	continuous	2	10
same_sentence(M_i,M_j)	0,1	1	0
same_paragraph(M_i,M_j)	0,1	1	0
subject_prev_agree(M_i,M_j)	0,1	0	0
subject_same_agree(M_i,M_j)	0,1	1	0
subject_closest_agree(M_i,M_j)	0,1	1	0
ssubject_prev_agree(M_i,M_j)	0,1	0	0
ssubject_same_agree(M_i,M_j)	0,1	1	0
ssubject_closest_agree(M_i,M_j)	0,1	1	0
closest_closest_agree(M_i,M_j)	0,1	0	0
closest_prev_agree(M_i,M_j)	0,1	0	0
closest_same_agree(M_i,M_j)	0,1	0	0
sfirst_prev_agree(M_i,M_j)	0,1	0	0
sfirst_same_agree(M_i,M_j)	0,1	1	0
sfirst_closest_agree(M_i,M_j)	0,1	1	0
pfirst_prev_agree(M_i,M_j)	0,1	0	0
pfirst_same_agree(M_i,M_j)	0,1	0	0
pfirst_closest_agree(M_i,M_j)	0,1	0	0
cb_closest_agree(M_i,M_j)	0,1	0	0
cb_prev_agree(M_i,M_j)	0,1	0	0
cb_same_agree(M_i,M_j)	0,1	0	0
subject_prev(M_i,M_j)	0,1	0	0
subject_same(M_i,M_j)	0,1	1	0
closest_prev(M_i,M_j)	0,1	0	0
closest_same(M_i,M_j)	0,1	0	0
sfirst_prev(M_i,M_j)	0,1	0	0
sfirst_same(M_i,M_j)	0,1	1	0
pfirst_prev(M_i,M_j)	0,1	0	0
pfirst_same(M_i,M_j)	0,1	0	0
ssubject_prev(M_i,M_j)	0,1	0	0
ssubject_same(M_i,M_j)	0,1	1	0
cb_prev(M_i,M_j)	0,1	0	0
cb_same(M_i,M_j)	0,1	0	0

Table 6.11: (continued)

Feature	Range	$pair_1$ value	$pair_2$ value
Pair's parameters, pronominal anaphors			
proana_subject_prev_agree(M_i,M_j)	0,1	0	0
proana_subject_same_agree(M_i,M_j)	0,1	1	0
proana_subject_closest_agree(M_i,M_j)	0,1	1	0
proana_ssubject_prev_agree(M_i,M_j)	0,1	0	0
proana_ssubject_same_agree(M_i,M_j)	0,1	1	0
proana_ssubject_closest_agree(M_i,M_j)	0,1	1	0
proana_closest_closest_agree(M_i,M_j)	0,1	0	0
proana_closest_prev_agree(M_i,M_j)	0,1	0	0
proana_closest_same_agree(M_i,M_j)	0,1	0	0
proana_sfirst_prev_agree(M_i,M_j)	0,1	0	0
proana_sfirst_same_agree(M_i,M_j)	0,1	1	0
proana_sfirst_closest_agree(M_i,M_j)	0,1	1	0
proana_pfirst_prev_agree(M_i,M_j)	0,1	0	0
proana_pfirst_same_agree(M_i,M_j)	0,1	0	0
proana_pfirst_closest_agree(M_i,M_j)	0,1	0	0
proana_cb_closest_agree(M_i,M_j)	0,1	0	0
proana_cb_prev_agree(M_i,M_j)	0,1	0	0
proana_cb_same_agree(M_i,M_j)	0,1	0	0
proana_subject_prev(M_i,M_j)	0,1	0	0
proana_subject_same(M_i,M_j)	0,1	1	0
proana_closest_prev(M_i,M_j)	0,1	0	0
proana_closest_same(M_i,M_j)	0,1	0	0
proana_sfirst_prev(M_i,M_j)	0,1	0	0
proana_sfirst_same(M_i,M_j)	0,1	1	0
proana_pfirst_prev(M_i,M_j)	0,1	0	0
proana_pfirst_same(M_i,M_j)	0,1	0	0
proana_ssubject_prev(M_i,M_j)	0,1	0	0
proana_ssubject_same(M_i,M_j)	0,1	1	0
proana_cb_prev(M_i,M_j)	0,1	0	0
proana_cb_same(M_i,M_j)	0,1	0	0

Baselines. As in the previous experiments, we use the SVM^{light}classifier with Soon et al.'s (2001) features as the main baseline. For the present experiment, we train the classifier on pairs with pronominal anaphors only. In all the tables in this section we show significant improvements over the baseline by */** and significant losses — by †/†† ($p < 0.05/p < 0.01$).

We have also implemented a set of naive salience baselines. For each pronoun, we first try to resolve it to the "most salient entity" of its clause. As the most salient entities for different baselines we consider CBs, subjects, sentence subjects, and first markables in a sentence/paragraph. If the most salient entity does not exist, it is the pronoun to be resolved, or it does not match the pronoun in person, number, or gender), we try to resolve it to the most salient entity of the previous sentence. If it is still not possible, we leave the pronoun unresolved.

Table 6.12 shows the performance of our baselines for pronominal anaphora resolution on the validation data. We see that the superficial features proposed by Soon et al. (2001) for full-scale coreference, cannot cope with pronominal anaphora. Most of the simple salience baselines show better performance, especially precision (χ^2-test, $p < 0.05$ for the "clause subject", "sentence subject" and "first in sentence" baselines). The "first in paragraph" and "CB" baselines have very low recall. This confirms the finding (see Section 6.6 for the statistics) that, although these entities are likely antecedents, they are not very common.

Overall, the best performing baseline is "pick the first NP in the current/previous sentence". It is also the most robust approach: virtually every sentence has a first NP that can be quickly and reliably computed.

Surprisingly, most existing approaches to pronoun resolution do not pay enough attention to this very simple baseline, opting for more complex solutions. A few studies rely on the order of mention partially: for example, ranking the CF list (Rambow, 1993) or influencing the order in which candidates are submitted to the resolution module (Kameyama, 1997; Harabagiu and Maiorano, 1999). Poesio (2003) shows that the first entity of the previous sentence is the best candidate for bridging anaphora resolution.

Throughout this experiment we will compare the system's performance to two baselines: an SVM classifier with the Soon et al.'s (2001) features, the baseline system employed throughout the thesis, and the "pick the first NP in the same/previous sentence" approach, our second baseline.

Features and Feature Groups. We have divided our salience feature set (Table 6.11) into three subgroups: main, recursive (shown in boldface) and secondary (shown in italic) features. The motivation and details of this distinction are explained in the beginning of this section.

To compute the values for the recursive features, we need a full-scale NP-

Baseline	Recall	Precision	F
SVM^{light}classifier, Soon et al's features	54.1	55.7	54.9
CB in same/previous sent.	††17.0	65.7	27.1
Clause subject in same/previous sent.	60.0	*66.4	63.0
Sentence subject in same/previous sent.	55.6	*68.2	61.2
First NP in same/previous sent.	60.0	*70.4	64.8
First-in-paragraph NP in same/previous sent.	†38.5	65.8	48.6

Table 6.12: Baselines for pronominal anaphora: performance on the validation data (3 "train" MUC-7 texts). Significant improvements over the main baselines (SVM^{light} learner, features of Soon et al. (2001)) are shown by */** and significant losses — by †/†† ($p < 0.05/p < 0.01$).

coreference module. In Sections 6.5 and 6.7 we have analyzed the distribution of recursive feature values in an ideal case — our training data annotated with all the required information. In this experiment, we use a fully automatic approach to obtain the values: we preprocess the validation set with Soon et al.'s (2001)-based SVM classifier (our general baseline system) and resolve all non-pronominal anaphors. The full-scale coreference resolution system of Soon et al. (2001) is one of the best existing algorithms for the task and, therefore, we hope to get a realistic picture of how accurately the recursive features can be extracted and how much the errors can affect pronoun resolution.

Table 6.13 shows the classifiers' performance for different feature groups. In all the cases, our approach significantly ($p < 0.05$) outperforms the first baseline (an SVM classifier with Soon et al.'s (2001) features) both in precision and recall. Compared to the second baseline, the precision gets slightly lower (not significant), but the recall is higher ($p < 0.05$ for the "main+recursive" feature set, not significant for other subgroups), resulting in a $4-6\%$ F-score gain.

The best result is obtained with the "main+recursive" group. This confirms the finding of Yang et al. (2004) that even imperfect full-scale coreference information helps to resolve pronominal anaphora.

Proximity and Sampling In Section 6.4 we have discussed two ways of encoding proximity information: explicitly, as features, or implicitly, by following the right-to-left first-link resolution strategy. We have mentioned that the implicit solution relies on a too local view of the problem and that the explicit solution might be incompatible with some sampling schemes. In the present experiment, we compare these two possibilities empirically.

We investigate two sampling/resolution strategies. The first one, originally proposed by Soon et al. (2001) for the full-scale NP-coreference task, works as follows: for the training data, each anaphor is paired with its closest antecedent

Feature groups	Recall	Precision	F
SVM^{light}classifier, Soon et al's features	54.1	55.7	54.9
First NP in same/previous sent.	60.0	*70.4	64.8
main	**71.1	67.1	69.1
main+recursive	**72.6	*69.5	71.0
main+secondary	**71.1	*68.6	69.8
main+recursive+secondary	*66.7	*70.9	68.7

Table 6.13: Performance on the validation data for different feature groups: "main" (the most relevant and robust), "recursive" (less robust), and "secondary" (less relevant) features. Significant improvements over the main baselines (SVM^{light} learner, features of Soon et al. (2001)) are shown by */** and significant losses — by †/†† ($p < 0.05/p < 0.01$).

(positive instance) and all the intervening markables (negative instances). For the testing data, each candidate anaphor is paired with all the preceding markables (candidate antecedents). They are submitted to the classifier one-by-one, starting from the rightmost candidate antecedent. Once a testing instance is classified as positive, the anaphor is linked to the corresponding antecedent and the resolution process stops. This strategy has originally been developed for the NP-coreference task. However, following its success for the full-scale problem, it has also been adopted by several approaches to pronominal anaphora (Yang et al., 2004).

The second sampling strategy, a window-based set-up, has been used in most traditional approaches to pronoun resolution. Each anaphor is paired with all preceding markables in an n-sentence window. At the testing stage, sentences are processed one-by-one from right to left. For each sentence, the best candidate is picked. If the system considers it plausible (for example, the classifier's confidence is above some threshold, no agreement constraints are violated, and so on), the anaphor is resolved to this candidate. Otherwise, the next sentence is processed. The window size is usually small and fixed. For the present experiment, we consider $n = 2$ and $n = 5$. In our training data, 91.8% of the anaphoric pronouns (corresponding to 80.9% of all the pronouns) have their antecedents in the same or preceding sentence ($n = 2$), and 98.6% (86.8%) – in the same or four preceding sentences ($n = 5$).

Table 6.14 shows the performance of our system for the different sampling strategies[13]. Both window-based schemes work worse than Soon et al. (2001) approach, with around 10% drop in precision (significant for $n = 5$, $p < 0.05$).

[13]Note that the first line refers not to the Soon et al. (2001) baseline (an SVM classifier with 12 "basic" features), but to an SVM classifier with salience features (main, recursive, and ±proximity), with the Soon et al. (2001) sampling strategy used to create training and testing instances.

Sampling strategy	No proximity features			Proximity features included		
	Recall	Precision	F	Recall	Precision	F
Soon et al.	72.6	69.5	71.0	72.6	69.5	71.0
5-sent. window	60.0	71.7	65.3	62.2	71.8	66.7
2-sent. window	63.7	71.7	67.5	65.2	72.1	68.5

Table 6.14: Performance on the validation data (3 MUC-7 "training" documents) for different sampling strategies, "main+recursive" feature set. No significant differences found between the left and the right parts.

Window size is important: the system shows better performance for the smaller window $n = 2$, both in precision and recall. This supports the position of linguistic theories of local coherence: pronominalization is highly influenced by very local discourse properties, in particular, by transitions between subsequent sentences. Consequently, a learning algorithm may achieve better performance for pronoun resolution by examining discourse clues in just pairs of adjacent sentences ($n = 2$) than by considering more distant candidates ($n = 5$).

Proximity features help slightly in the case of window-based sampling: we see a minor improvement in precision and recall for both $n = 2$ and $n = 5$ (not significant). For the Soon et al. (2001) sampling scheme, proximity features do not affect the performance at all: the corresponding right-to-left first-link resolution strategy strongly favors recent candidates, making proximity features redundant.

Resolving Different Pronouns Different pronouns may require different resolution strategies. For example, the interpretation of third person pronouns usually depends on the *discourse context* — a few preceding sentences. First and second person anaphors, on the contrary, rely on the *situational context* — the identities of the speaker and the hearer.

Most state-of-the-art systems for pronominal anaphora resolution concentrate on the third person. In this study, we rely on a uniform strategy for all pronouns. However, some of them are more difficult or less suitable for our approach. We have evaluated the algorithm's performance for different pronouns[14] to identify and analyze problematic cases. Table 6.15 shows the performance figures for different groups of pronouns.

Personal and possessive pronouns can be resolved equally well, although they are very different in their nature: almost all possessive pronouns have their antecedents in the same clause, whereas personal pronouns (especially

[14] All the pronouns were used for training. The (same) obtained classifier was then tested on different pronouns in the validation set.

subjects) tend to have their antecedents in the preceding discourse. Reflexive pronouns are extremely rare: we have not found a single reflexive pronoun in our validation data.

Most pronouns belong to the "third person" group. Our salience-based algorithm, as expected, can resolve them more or less reliably. First and second person pronouns should be more problematic — their resolution involves more complex inference, not supported by our features. Nevertheless, the classifier performs surprisingly well on 1st person pronouns, achieving an F-score of 73.0%. It means that this group, around 18% of all the pronouns in newswire documents, can be analyzed based on contextual clues and therefore, contrary to the existing practice (ranging from the first (Hobbs, 1978) to the most recent approaches (Yang et al., 2004)), there is no need to exclude first person pronouns from the scope of computational approaches to pronominal anaphora.

Second person pronouns are genuinely difficult — neither recall nor precision achieve a reasonable level. We have identified several potential problems. First, second person pronouns often have referents not explicitly mentioned or connected to their discourse segments:

(83) "We secure all our cargo," said [the official]$_1$, who said most American carriers in Greece have their own cargo X-ray machines. "[I]$_{2,ante=1}$ can't tell [you]$_3$ we X-ray everything..."

(84) "I caught [Reggie]$_1$ when he was much younger counting his dad's trophies," [McNair]$_2$ said. "And [I]$_{3,ante=2}$ said, 'Well, hey, what are [you]$_{4,ante=1}$ doing?'

If we could correctly identify embedded subdiscources as produced by "the official" and "McNair", we could resolve "I". But even in such a case, we don't have enough information to link "you" in example (83) to the journalist who has interviewed the official. Resolution of "you" to "Reggie" in example (84) relies on a complex inference scheme: linking together "catching" and "saying" actions to determine that they have, or might have, the same patient. As these examples suggest, identifying the hearer is a much more difficult task than identifying the speaker.

Second, many second person pronouns are not anaphoric:

(85) Boeing's antenna "allows [you] to capture a signal on the move," said Rich Williams, project manager for Boeing Defense & Space Group's Phased Array Systems Organization.

There have been several studies on identifying non-anaphoric "it" (see Chapter 7 for discussion), but we are not aware of any computational approach to identifying non-anaphoric "you".

Pronoun groups	Recall		Precision		F
All pronouns	98/135	72.6%	98/141	69.5.0%	71.0
Personal	69/98	70.4%	69/103	67.0%	68.7
Reflexive	0/0	-	0/0	-	-
Possessive	26/37	70.3%	26/38	68.4%	69.3
1st person	19/25	76.0%	19/27	70.3%	73.0
2nd person	1/9	11.1%	1/9	11.1%	11.1
3rd person	72/101	71.3%	72/105	68.6%	69.9
Singular	69/93	74.2%	69/95	72.6%	73.4
Plural	26/33	78.8%	26/37	70.3%	74.3
Ambiguous	1/9	11.1%	1/9	11.1%	11.1

Table 6.15: Performance on the validation data (3 MUC-7 "training" documents) for different groups of pronouns, "main+recursive" feature set.

Third, English second person pronouns are ambiguous in number. Therefore we cannot apply agreement constraints to filter out potential candidates and reduce the search space.

All these factors make second person pronoun resolution in written texts a challenging problem, not yet addressed extensively in the literature.

The number factor might also affect the resolution quality: plural pronouns can refer to group entities (see Section 4.9 for the number disagreement statistics) or conjoined NPs that are difficult to extract reliably from an error-prone parser. Nevertheless, the classifier shows the same performance level for singular and plural pronouns. The problematic "ambiguous" group consists of second person pronouns discussed above.

Comparison to Related Work on Pronoun Resolution We have seen that our salience-based machine learning approach outperforms both the relatively knowledge-poor algorithm of Soon et al. (2001) and naive salience baselines. However, the former was not designed for pronominal anaphora and the latter ones are too simple. In order to get a realistic assessment of our approach, we have to evaluate it against state-of-the-art algorithms in pronoun resolution.

Numerous pronoun resolution algorithms, symbolic and statistic, have been proposed in the literature (Hobbs 1978; Brennan et al. 1987; Lapin and Leass 1994; Kennedy and Boguraev 1996; Ge et al. 1998; Mitkov 1998; Strube 1998; Tetreault 2001; Yang et al. 2004, among others). Unfortunately, they are evaluated on different corpora, which makes direct comparison to our approach difficult: the experiments of Tetreault (2001) confirm that, for many anaphora resolution algorithms, performance figures are strongly domain- and corpus-

dependent. For the present experiment, we have chosen a family of algorithms proposed by Yang et al. (2004) — the only approach we are aware of, evaluated on the MUC-7 corpus. This is a very recent study, reflecting the latest advances in pronoun resolution.

Yang et al. (2004) present a machine learning approach. They use 14 "baseline" features, encoding different kinds of knowledge, and 9 "backward" features. Their backward features roughly correspond to the subset of our "recursive" features describing "Candidate's antecedent" (Section 6.7). Yang et al. (2004) investigate several ways to obtain values for the backward features in a real-world application, introducing the RealResolve1–RealResolve4 algorithms. The RealResolve1 version corresponds to our way of extracting recursive features.

Yang et al. (2004) only attempt to resolve third person anaphoric pronouns. For each anaphor, a set of candidates is obtained with the Soon et al. (2001) sampling strategy. It is then filtered with agreement constraints. The remaining pairs are described as feature vectors and used by the C5.0 decision tree induction system (Quinlan, 1993) to learn a classifier.

To allow a fair comparison, we have slightly modified our framework. We have discarded all the first and second person and non-anaphoric pronouns from both the training and testing data to have the same range of covered phenomena. We do not use any prefiltering: in the present experiment, we are only interested in the impact of salience on pronoun resolution, so, we do not rely on any additional knowledge.

Table 6.16 compares our approach to Yang et al.'s (2004) on the MUC-7 test data (there are no validation data figures for Yang et al. (2004) available). Without recursive features, the algorithm of Yang et al. (2004) shows significantly higher recall ($p < 0.05$) and slightly lower precision. With the recursive features computed in the same way (RealResolve1), our algorithm shows a slight gain in recall, but loses in precision (not significant). More sophisticated strategies for extracting recursive features, RealResolve3 and RealResolve4, significantly outperform our approach in recall with a non-significant drop in precision.

The main problem of our approach is recall. This is not surprising: with purely salience-based features, we cannot hope to account for all cases of pronominalization.[15] Correspondingly, our approach shows lower recall figures compared to Yang et al. (2004). Nevertheless, at least in some settings, the difference is not significant.

Summarizing, our pure salience-based approach shows acceptable performance figures, comparable to the state-of-the-art. This makes it a good basis for a pronoun resolution system. But salience alone is obviously not enough —

[15]For example, in Section 4.6, we have seen how syntactic factors may affect the interpretation of pronouns.

Features	Recall	Precision	F
Our approach			
main	60.7	69.2	64.7
main+recursive	63.2	72.0	67.3
Yang et al's approach			
No recursive features	71.9	68.6	70.2
RealResolve1	62.3	77.7	69.1
RealResolve2	63.0	77.9	69.7
RealResolve3	74.7	67.3	70.8
RealResolve4	74.7	67.3	70.8

Table 6.16: Our salience-based approach compared to Yang et al's (2004) algorithms: performance on anaphoric 3rd person pronouns in the test data (20 MUC-7 "formal test" documents). Significance tests not applicable.

by adding another kinds of knowledge, one could handle more cases of pronominalization, thus, improving the algorithm's recall. We will come back to this point in Chapter 8, where we combine salience with other knowledge types.

6.8.2 Experiment 8: Salience-based NP-Coreference Resolution

In this experiment we evaluate the impact of discourse and salience features on full-scale NP-coreference resolution. We train an SVM^{light}classifier on the 30 MUC-7 "dry-run" documents and test it on the test data, 20 "formal test" documents, and the validation data, 3 "formal train" documents. The sampling strategy of Soon et al. (2001) is used to generate training and testing instances from the corpus annotation. As a naive baseline, we merge all the entities to form one chain. We also train an SVM^{light}classifier with Soon et al.'s (2001) features as a second, more sophisticated baseline.

Pure salience-based approach. Table 6.17 shows the system's performance for the "main" and "main+recursive" feature subgroups. We see that salience alone is a very bad predictor for full-scale coreference, achieving low performance. This can be explained by taking a closer look at the nature of salience features: they aim at picking the candidates that are *generally* very likely antecedents, without checking whether a candidate is a good antecedent for a particular anaphor. Consequently, most anaphors are resolved to just a few candidates from the very beginning of the document, forming one big chain — in Section 6.2 we have seen that markables from the SLUG and DATE sections are very likely antecedents.

Comparing to other knowledge types, the pure salience-based approach performs poorly. We see that, for example, syntactic information allows us

	Test set			Validation set		
Features	Recall	Prec.	F	Recall	Prec.	F
Baselines						
"merge all"	**86.6	††35.2	50.0	**91.9	††38.0	53.7
Soon et al.	50.5	75.3	60.4	54.5	56.9	55.7
Salience features						
main	86.6	35.2	50.0	91.9	38.0	53.7
main+recursive	**86.6	††35.2	50.0	**91.9	††38.0	53.7
Other knowledge sources						
matching	52.2	††61.2	56.3	56.2	53.3	54.7
syntax	††7.6	68.5	13.8	††9.9	57.4	16.9
semantics	††28.5	††48.3	35.9	††37.9	†48.5	42.6

Table 6.17: A salience-based approach to the full-scale Coreference Resolution task: performance on the testing (20 MUC-7 "formal test" documents) and the validation (3 MUC-7 "train" documents) data. Significant improvements over the main baselines (SVM^{light} learner, features of Soon et al. (2001)) are shown by */** and significant losses — by †/†† ($p < 0.05/p < 0.01$).

to resolve very few anaphors, but this can be done reliably (with a precision of 68.5%). A salience-based classifier merges all the entities into one chain, proposing an uninformative partition of the markables into coreference classes.

Summarizing, we have seen that salience is important for pronominal anaphora (Experiment 7), but it does not provide enough evidence for full-scale coreference resolution.

Combining Salience with the Basic Coreference Features. Discourse knowledge alone can be misleading: although a salience-based algorithm can estimate the probability of a candidate to be an antecedent in general, we have no way to determine whether it is a good antecedent for a given anaphor. Obviously, we need other kinds of information to assess the "fit" between two markables. We have combined our salience-based features with the "basic" features proposed by Soon et al. (2001). Basic subset contains just 12 superficial features, representing different knowledge types.

The performance of the mixed "basic+salience" approach is shown in Table 6.18. For comparison, we also show the performance of the mixed "basic+ matching" (Chapter 3), "basic+syntax" (Chapter 4), and "basic+semantics" (Chapter 5) classifiers.

We see that, combined with the basic features, our salience-based algorithm shows a significant gain in recall and a slight gain in precision over the baseline. It yields an F-score of 63.7-63.8% for the MUC-7 test corpus, which is a very

Features	Test set Recall	Prec.	F	Validation set Recall	Prec.	F
Baselines						
"merge all"	86.6	35.2	50.0	91.9	38.0	53.7
Soon et al.	50.5	75.3	60.4	54.5	56.9	55.7
Salience+basic features						
basic+main	**55.4	75.3	63.8	*63.1	63.1	63.1
basic+main+recurs.	*54.7	76.0	63.7	*62.3	62.7	62.5
Other knowledge sources						
basic+matching	**58.4	††63.1	60.6	60.6	54.1	57.1
basic+syntax	52.2	75.2	61.7	56.2	56.5	56.4
basic+semantics	**55.6	††67.5	61.0	60.3	55.1	57.6

Table 6.18: Basic coreference resolution algorithm augmented with discourse knowledge, performance on the testing (20 MUC-7 "formal test" documents) and the validation (3 MUC-7 "train" documents) data. Significant improvements over the main baselines (SVM^{light} learner, features of Soon et al. (2001)) are shown by */** and significant losses — by †/†† ($p < 0.05/p < 0.01$).

promising result, compared to other systems evaluated on this dataset (only Ng and Cardie (2002c) report better figures).

Although adding other kinds of knowledge to the baseline classifier results in an F-score improvement as well, a salience-based approach is the most promising. It is also the only one showing statistically significant improvement on the validation data.

6.9 Summary

In this chapter we have investigated the influence of discourse and salience factors on coreference resolution: document structure (its composition and linear organization, Section 6.2), discourse levels (quoted opinions of other speakers, Section 6.3), proximity (Section 6.4), properties motivated by Centering theory (Section 6.5), and the coreferential status of candidate antecedents (Section 6.7).

The distribution of markables and the anaphoric links between them suggests that salience factors significantly affect coreference and are therefore potentially relevant for our task. We have seen, however, that discourse factors are more important for pronouns than for other types of anaphors: 80.9% of pronominal anaphors have their antecedents in the pool of "salient candidates", compared to 43.3% on average (Section 6.6).

We have encoded discourse and salience knowledge in 97 features listed

in Table 6.11 and trained an SVM^{light}classifier with these features to build a salience-based coreference resolution system. It has been tested on two tasks: pronominal anaphora (Experiment 7, Section 6.8.1) and full-scale NP-coreference (Experiment 8, Section 6.8.2).

For pronominal anaphora, we observed a promising performance level: our pure discourse-based approach outperforms the baseline motivated by Soon et al. (2001) and naive salience baselines. We have evaluated the impact of different feature subsets and sampling strategies on the algorithm's performance, as well as its quality for different classes of pronouns.

For full-scale NP coreference resolution, we have seen that salience alone is a bad predictor: the system produces an uninformative partition, merging all markables into a single chain. However, augmented with very superficial features of Soon et al. (2001), our salience-based approach achieves a performance level of 63.8% — the best up-to-date result for the MUC-7 data.

To summarize, our experiments show again the importance of combining different knowledge types: coreference is a complex phenomenon involving several linguistic factors. In Chapter 8 we will investigate a multi-factor approach, incorporating all knowledge sources investigated so far.

Chapter 7

Anaphoricity and Antecedenthood

In the previous chapters we have investigated numerous linguistic factors important for coreference. They help a coreference resolution engine decide on the plausibility of an anaphoric link between two markables. Not all NPs in a document, however, participate in coreference relations, and, even if they do, they often can only be anaphors or antecedents, but not both. In the present chapter we will investigate possibilities to automatically reduce the pool of anaphors and antecedents by filtering out unlikely candidates.

In some cases, we can determine if a markable could potentially be an anaphor or an antecedent by looking at its structure and surrounding context. Consider the following example:

(86) Shares in [Loral Space]$_1$ will be distributed to Loral shareholders. [The new company]$_{2,ante=1}$ will start life with [no debt]$_3$ and $700 million in cash. [Globalstar]$_4$ still needs to raise [$600 million]$_5$, and Schwartz said that [the company]$_{6,ante=4}$ would try to raise [the money]$_{7,ante=5}$ in [the debt market]$_8$.

In (86), the third markable, "no debt" can be neither an anaphor, nor an antecedent. We can tell that by looking at its structure — with the determiner "no", this description does not refer to any entity. The second, sixth and seventh markables are all definite descriptions and therefore are likely to be anaphoric. Although the eighth markable, "the debt market" is a definite NP as well, it is a uniquely referring description and thus it might as well be non-anaphoric. Finally, the fifth markable, "$600 million" is a possible antecedent (and is indeed mentioned again as "the money" later), but not a very likely anaphor.

Most coreference resolution systems, including, for example, the algorithm of Soon et al. (2001) try to resolve *all* "candidate anaphors" by comparing them to all preceding "candidate antecedents" until the correct one is found. Such approaches require substantial amount of processing: in the worst case one has to check $\frac{n(n-1)}{2}$ candidate pairs, where n is the total number of markables found by the system. Moreover, spurious coreference links may appear when, for example, a non-anaphoric description is resolved to some preceding markable.

Poesio et al. (1997) have shown that such an exhaustive search is not needed, because many noun phrases are not anaphoric at all: more than 50% of definite NPs in their corpus have no prior referents. Obviously, this number is even higher if one takes into account all the other types of NPs — for example, only around 30% of our (automatically extracted) markables are anaphoric.

We can conclude that a coreference resolution engine might benefit from a pre-filtering algorithm for identifying non-anaphoric and non-antecedent descriptions. First, we save much processing time by discarding at least half of the markables. Second, the prefiltering module is expected to improve the system's precision by discarding spurious candidates.

In Section 7.1 we briefly summarize theoretical research on anaphoricity and referentiality and discuss the related applications. Note that theoretical studies focus on referentiality, whereas we will consider a related task of detecting antecedenthood (this will be described in details below). In Section 7.2 we experiment on learning anaphoricity and antecedenthood filters from the MUC data. We also incorporate the anaphoricity and antecedenthood classifiers into a baseline no-prefiltering coreference resolution system to see if such prefiltering modules help.

7.1 Related Work

In this section, we present an overview of theoretical studies of referentiality (Karttunen, 1976) and anaphoricity (Prince, 1981). We also discuss relevant computational approaches (Bean and Riloff, 1999; Vieira and Poesio, 2000; Ng and Cardie, 2002b; Uryupina, 2003a; Poesio et al., 2004; Byron and Gegg-Harrison, 2004).

Karttunen (1976) points out that in some cases an NP, in particular an indefinite one, does not refer to any entity:

(87) Bill doesn't have [a car].

Obviously, (87) does not imply the existence of any specific "car". In Karttunen's (1976) terms, the NP "a car" does not establish a *discourse referent* and therefore it cannot participate in any coreference chain — none of the alternatives in (88) can follow (87):

(88) A. [It] is black.
B. [The car] is black.
C. [Bill's car] is black.

Karttunen (1976) identifies several factors affecting referential status of NPs, including modality (89), negation (90), or nonfactive verbs (91):

(89) Bill can make [a kite]. *[The kite] has a long string.

(90) Bill didn't dare to ask [a question]. *The lecturer answered [it].

(91) I doubt that Mary has [a car]. *Bill has seen [it].

Karttunen (1976) gives more examples of referential and non-referential NPs, showing that an extensive analysis of the phenomenon requires sophisticated inference: "In order to decide whether or not a nonspecific indefinite NP is to be associated with a referent, a text-interpreting device must be able to assign a truth value to the proposition represented by the sentence in which the NP appears. It must be sensitive to the semantic properties of verbs that take sentential complements; distinguish between assertion, implication, and presupposition; and finally, it must distinguish what exists for the speaker from what exists only for somebody else".

Byron and Gegg-Harrison (2004) present an algorithm for identifying "non-licensing" NPs based on Karttunen's (1976) theory of referentiality. Their approach relies on a hand-crafted heuristic, encoding some of Karttunen's (1976) factors (for example, an NP is considered non-referential if its determiner is "no"). In the present study we represent this information as features for machine learning.

Numerous theories of anaphoricity, especially for definite descriptions, have been proposed in the literature (Hawkins 1978; Prince 1981; Loebner 1985; Fraurud 1990, among others). We point the reader to Vieira (1998) for an extensive overview and comparison of the major theoretic studies in the field.

The theories aim at interpreting (definite) descriptions by relating them to the linguistic and situational context and, more specifically, to their antecedents. From this perspective, an NP may be *given* (related to the preceding discourse) or *new* (introducing an independent entity). The theories of anaphoricity provide different detailed subclassifications of given and new descriptions. For example, Prince (1981) distinguishes between the discourse and the hearer givenness. This results in the following taxonomy:

- *brand new* NPs introduce entities which are both discourse and hearer new ("a bus"), some of them, *brand new anchored* NPs, contain explicit link to some given discourse entity ("a guy I work with"),

- *unused* NPs introduce discourse new, but hearer old entities ("Noam Chomsky"),

- *evoked* NPs introduce entities already present in the discourse model and thus discourse and hearer old: *textually evoked* NPs refer to entities which have already been mentioned in the previous discourse ("he" in "A guy I worked with says he knows your sister"), whereas *situationally evoked* are known for situational reasons ("you" in "Would you have change of a quarter?"),

- *inferrables* are not discourse or hearer old, however, the speaker assumes the hearer can infer them via logical reasoning from evoked entities or other inferrables ("the driver"in "I got on a bus yesterday and the driver was drunk"), *containing inferrables* make this inference link explicit ("one of these eggs").

Linguistic theories, including (Prince, 1981), focus on anaphoric usages of definite descriptions (either evoked or inferrables). Recent corpus studies (Poesio and Vieira, 1998) have revealed, however, that more than 50% of (definite) NPs in newswire texts are not anaphoric. These findings have motivated recent approaches to automatic identification of *discourse new* vs. *discourse old* NPs.

Several algorithms for identifying discourse-new markables have been proposed in the literature. Vieira and Poesio (2000) use hand-crafted heuristics, encoding syntactic information. For example, the noun phrase "the inequities of the current land-ownership system" is classified by their system as $+$*discourse_new*, because it contains the restrictive postmodification "of the current land-ownership system". This approach leads to 72% precision and 69% recall for definite discourse-new NPs on Vieira and Poesio's (2000) corpus.

Bean and Riloff (1999) make use of syntactic heuristics, but also mine additional patterns for discourse-new markables from corpus data. They consider four types of non-anaphoric markables:

1. having specific syntactic structure,

2. appearing in the first sentence of some text in the training corpus,

3. exhibiting the same pattern as several expressions of type (2),

4. appearing in the corpus at least 5 times and always with the definite article (*"definites-only"*).

Using various combinations of these methods, Bean and Riloff (1999) achieve an F-measure for *existential* NPs of about $81-82\%$ on the MUC-4 data.[1] The algorithm, however, has two limitations. First, one needs a corpus consisting of many small texts. Otherwise it is impossible to find enough non-anaphoric markables of type (2) and, hence, to collect enough patterns for the markables of type (3). Second, for an entity to be recognized as "definite-only", it should be found in the corpus at least 5 times. This automatically results in a data sparseness problem, excluding many infrequent nouns and NPs.

In an earlier paper (Uryupina, 2003a) we have proposed a web-based algorithm for identifying discourse-new and unique NPs. Our approach helps overcome the data sparseness problem of Bean and Riloff (1999) by relying on Internet counts. Unfortunately, this methodology cannot be used at the moment, since major Internet search engines, for example, AltaVista, currently treat articles "a", "an", and "the" as too frequent words not affecting search queries. This makes it no longer possible to compare cooccurence statistics for different types of determiners.

The above-mentioned algorithms for automatic detection of discourse-new and non-referential descriptions are helpful for interpreting NPs, accounting for documents' information structure. From a generation perspective, this knowledge can be helpful for article generation (see, for example, Minnen et al. (2000)). However, it is not a priori clear whether such approaches are useful for coreference resolution. On the one hand, discarding discourse-new and/or non-referential NPs from the pool of candidate anaphors and antecedents, we can drastically narrow down the algorithm's search space. This reduces the processing time and makes candidate re-ranking much easier. On the other hand, errors, introduced by automatic anaphoricity or referentiality detectors, may propagate and thus deteriorate the performance of a coreference resolution engine.

Ng and Cardie (2002b) have shown that an automatically induced detector of non-anaphoric descriptions leads to performance losses for their coreference resolution engine, because too many anaphors are misclassified as discourse-new. To deal with the problem, they have augmented their discourse-new classifier with several precision-improving heuristics. In our web-based study (Uryupina, 2003a) we have tuned machine learning parameters to obtain a classifier with a better precision level. In a later study, Ng (2004) relies on held-out data to optimize relevant learning parameters and to decide on the possible system architecture.

Byron and Gegg-Harrison (2004) report ambivalent results concerning the importance of a referentiality detector for coreference. On the one hand, the incorporation of referentiality prefiltering in several pronoun resolution algo-

[1] Bean and Riloff's (1999) *existential* class contains not only *brand new* NPs, but also all mentions (including anaphoric) of unique description, such as "the pope" or "the FBI".

rithms does not yield any significant precision gains. On the other hand, such a prefiltering significantly reduced the systems' processing time. It must be noted that Byron and Gegg-Harrison (2004) have tested their referentiality detector only for pronominal anaphora. We are not aware of any approach integrating referentiality into a full-scale coreference resolution system.

To summarize, several algorithms for detecting non-referring or non-anaphoric descriptions have been proposed in the literature. These studies revealed two major problems. First, it is necessary to identify and represent relevant linguistic factors affecting the referentiality or anaphoricity status of an NP. Second, incorporating error-prone automatic modules for identifying discourse-new or non-referential descriptions into a coreference resolution engine is a non-trivial task of its own: when not properly optimized, such modules may lead to performance losses. We will address these two problems in the following section (Experiments 9 and 10).

7.2 Experiments

7.2.1 Experiment 9: Identifying Non-anaphors and Non-antecedents

The corpus studies of Poesio and Vieira (1998) suggest that human annotators are able to successfully distinguish between anaphoric (discourse old) and non-anaphoric (discourse-new) descriptions. This motivates the present experiment: using machine learning techniques we try to automatically detect probable anaphors and antecedents. In our next experiment (Section 7.2.2) we will incorporate our anaphoricity and referentiality classifiers into a coreference resolution system.

Data. Throughout this thesis we use the same dataset, the MUC-7 corpus. In all preceding experiments, we have relied on the coreference chains annotations of the MUC data. In the present experiment, however, we need two other kinds of information — anaphoricity and referentiality. This knowledge can to some extent be inferred from the coreference data as provided by MUC.

We have automatically annotated our NPs as $\pm$*discourse_new* using the following simple rule: an NP is considered $-$*discourse_new* if and only if it is marked in the original MUC-7 corpus and has an antecedent. Our $+$*discourse_new* class corresponds to the union of Prince's *brand new*, *unused*, *situationally evoked*, and *inferrable* groups. Our $-$*discourse_new* (i.e. , *discourse_old*) corresponds to Prince's *textually evoked*. Poesio and Vieira (1998) have shown that around 20% of (definite) descriptions in their corpus are inferrables. They also report low inter-annotator statistics for fine-grained NP-classification schemes (e.g., the one of Prince (1981)), showing that humans have difficulties distinguishing between *brand_new/unused*, *situationally_evoked*, and *inferrables*. This makes the simple $\pm$*discourse_new* classifica-

tion preferable, as long as we do not attempt to propose bridging antecedents (i.e., resolve inferrables).

Extracting referentiality information from coreference annotated data is by far less trivial. By definition (Karttunen, 1976), non-referential descriptions cannot be antecedents for any subsequent NPs. Consider, however, the following example:

(92) There was [no listing]$_1$ for [the company]$_2$ in [Wilmington]$_3$.

In (92), the NP "no listing" is not referential and, therefore, cannot be an antecedent for any subsequent markable. Both "the company" and "Wilmington", on the contrary, are referential and could potentially be re-mentioned. However, this does not happen, as the document ends with the next sentence. By looking at coreference annotated data, we can only say whether an NP is an antecedent, but, if it is not, we cannot decide if it is referential (as "the company" or "Wilmington") or not (as "no listing"). Consequently, we cannot automatically induce referentiality annotation from coreference data.

For our main task, coreference resolution, we are not exactly interested in the referential vs. non-referential distinction. We would rather like to know how likely it is for a markable to be an antecedent. Therefore, instead of a referentiality detector in the strict sense, we need a $\pm ante$ labelling: an NP is considered $+ante$, if it is annotated in MUC-7 and is an antecedent for some subsequent markable. We have therefore changed the scope of the present experiment to detecting antecedenthood — the probability for a markable to be an antecedent.

In the present experiment, we rely on 30 MUC-7 "dry-run" documents for training. For testing, we use our validation (3 MUC-7 "train" documents) and testing (20 MUC-7 "formal test" documents) sets. This results in 5028 noun phrases for training and 976/3375 for the validation/testing data. 3325 training instances were annotated as $+discourse\_new/-ante$ and 1703 — as $-discourse\_new/+ante$[2] (613/2245 and 363/1130 for testing). All the performance figures reported below are for $+discourse\_new$ and $-ante$ classes.

Features. Table 7.1 shows the features we have used in our experiment and their values for the "no listing" and "the company" markables in Example (92). They can be divided into the following groups: surface, syntactic, semantic, salience, same-head, and Karttunen's (1976) factors.

The former four groups have already been discussed above: we have re-used all the unary features introduced in Chapters 3–6 (see Tables 3.8, 4.15, 5.8, and 6.11).

[2] As each anaphor is linked to exactly one antecedent according to the MUC-7 annotation guidelines, there is a one-to-one correspondence between $-discourse\_old$ and $+ante$ classes.

"Same-head" features represent coreference knowledge on a very simplistic level. The boolean feature `same_head_exists` shows if there exists a markable in the preceding discourse with the same head as the given NP, and the continuous feature `same_head_distance` encodes the distance to this markable. Obtaining values for these features does not require exhaustive search when heads are stored in an appropriate data structure, for example, in a trie. The motivation for "same-head" features comes from Vieira and Poesio (2000) and Poesio et al. (2004): they clearly show that anaphoricity detectors might significantly benefit from an early inclusion of a simplified coreference check.

Table 7.1: Features for anaphoricity and antecedenthood detectors and their values for the markables "no listing"$_1$ and "the company"$_2$ in Example (92). Surface, syntactic, semantic and salience-based features have been introduced in Tables 3.8, 4.15, 5.8, and 6.11 above.

Features	Range	NP_1 value	NP_2 value
Surface features			
length_s(M_i)	continuous	10	11
length_w(M_i)	continuous	2	2
digits(M_i)	0,1	0	0
alphas(M_i)	0,1	1	1
lower_case(M_i)	0,1	1	1
upper_case(M_i)	0,1	0	0
cap_words(M_i)	0,1	0	0
digits_h(M_i)	0,1	0	0
alphas_h(M_i)	0,1	1	1
lower_case_h(M_i)	0,1	0	0
upper_case_h(M_i)	0,1	1	1
cap_words_h(M_i)	0,1	0	0
Syntactic features			
type_of_markable(M_i)	nominal	OTHER	DEFNP
type_of_pronoun(M_i)	nominal	NONE	NONE
type_of_definite(M_i)	nominal	NONE	THE
determiner(M_i)	nominal	no	the
det_ana_type(M_i)	nominal	DET_nonana	DET_ana
det_ante_type(M_i)	nominal	DET_nonante	DET_ante
head_anaphoric(M_i)	0,1	0	1
head_nonanaphoric(M_i)	0,1	0	0
head_antecedent(M_i)	0,1	0	0
head_nonantecedent(M_i)	0,1	0	0
coordination(M_i)	0,1	0	0
premodified(M_i)	0,1	0	0
postmodified(M_i)	0,1	1	0
postrestrictive(M_i)	0,1	0	0
grammatical_role(M_i)	nominal	OBJ	PPCOMPL
subject(M_i)	0,1	0	0
sentence_subject(M_i)	0,1	0	0
minimal_depth_subject(M_i)	0,1	0	0
number(M_i)	nominal	SG	SG
person(M_i)	nominal	3	3

Table 7.1: (continued)

Features	Range	NP_1 value	NP_2 value
Semantic features			
semclass_ne(M_i)	nominal	OBJECT	ORG
semclass_soon(M_i)	nominal	OBJECT	ORG
gender(M_i)	nominal	OBJECT	OBJECT
Salience features			
section_tag(M_i)	nominal	TEXT	TEXT
paragraph_number_bin(M_i)	1...10	10	10
sentence_number_bin(M_i)	1...10	10	10
paragraph_rank_bin(M_i)	1...10	9	10
sentence_rank_bin(M_i)	1...10	3	6
embedded(M_i)	0,1	0	0
cb(M_i)	0,1	0	0
subject(M_i)	0,1	0	0
sentence_subject(M_i)	0,1	0	0
first_in_sentence(M_i)	0,1	0	0
first_in_paragraph(M_i)	0,1	0	0
"Same-head" features			
same_head_exists(M_i)	0,1	0	0
same_head_distance(M_i)	continuous	-	-
Karttunen (1976)-motivated features			
part_of_apposition(M_i)	1st, 2nd, none	NONE	NONE
predicative_NP(M_i)	0,1	0	0
in_negated_clause(M_i)	0,1	0	0
obj_in_modal_clause(M_i)	0,1	0	0
grammatical_role(M_i)	nominal	OBJ	PPCOMPL
determiner(M_i)	nominal	no	the
semclass_ne(M_i)	nominal	OBJECT	ORG

The last group encodes the referentiality-related factors investigated by Karttunen (1976) and Byron and Gegg-Harrison (2004): apposition, copula, negation, modal constructions, determiner, grammatical role (especially, modifier), and semantic class (especially, MONEY, PERCENTAGE, etc). The values are extracted from a parser's and an NE-tagger's output. Some of Karttunen's (1976) features also belong to other groups: `grammatical_role`, `determiner` (also syntactic), and `semantic_class` (also semantic).

Altogether we have 49 features: 12 surface, 20 syntactic, 3 semantic, 10 salience, 2 "same-head", and 7 of Karttunen's (1976) constructions, corresponding to 123 boolean/continuous features.

Features	Test data			Validation data		
	Recall	Precision	F	Recall	Precision	F
Baseline	100	66.52	79.89	100	62.81	77.16
All	††93.54	**82.29	87.56	††80.26	**84.68	82.41
Surface	100	66.52	79.89	100	62.81	77.16
Syntactic	††97.37	**71.96	82.76	††93.64	**72.57	81.77
Semantic	††98.53	*68.89	81.09	††97.06	**69.11	80.73
Salience	††91.22	*69.26	78.74	††89.23	65.98	75.86
Same-Head	††84.45	**81.16	82.77	††76.35	**82.11	79.13
Karttunen's	††91.63	**71.15	80.10	††88.58	66.71	76.10
Synt+SH	††89.98	**83.51	86.62	††78.96	**85.06	81.90

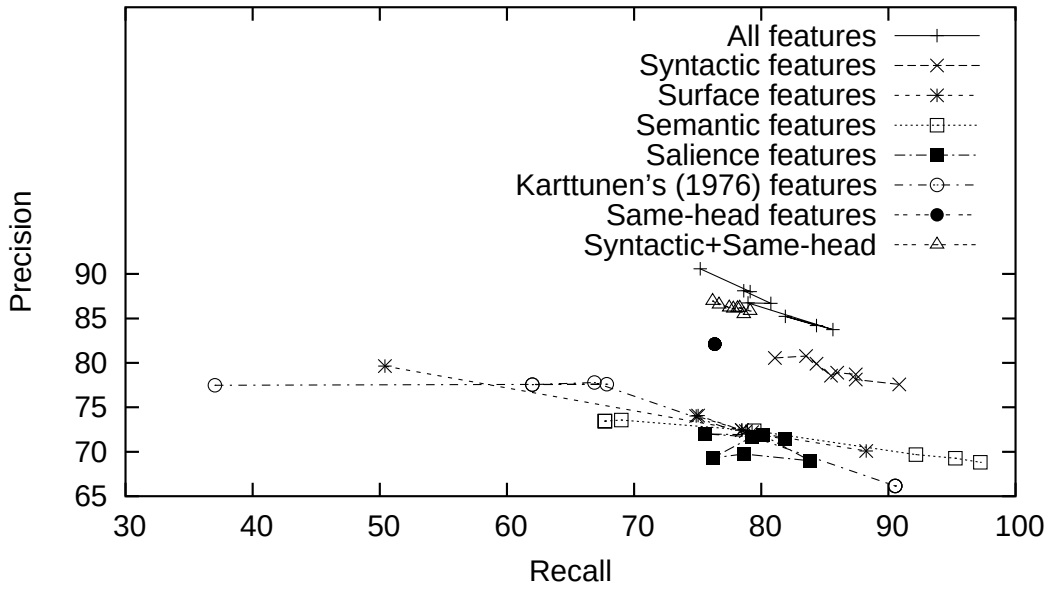
Table 7.2: An SVM-based anaphoricity detector: performance for the +*discourse_new* class on the validation (3 MUC-7 "train" documents) and test (20 MUC-7 "formal" documents) data.

Identifying discourse-new markables. As a baseline for our experiments we use the major class labelling: all markables are classified as +*discourse_new*. This results in F-scores of 79.9% and 77.2% for the testing and validation data. This baseline can be used as a comparison point for ±*discourse_new* detectors. However, it has no practical relevance for our main task, coreference resolution: if we classify all the markables as +*discourse_new* and, consequently, discard them, the system would not even try to resolve any anaphors. In all the tables in this Section we show significant improvements over the baseline for $p < 0.05/p < 0.01$ by */** and significant losses — by †/††.

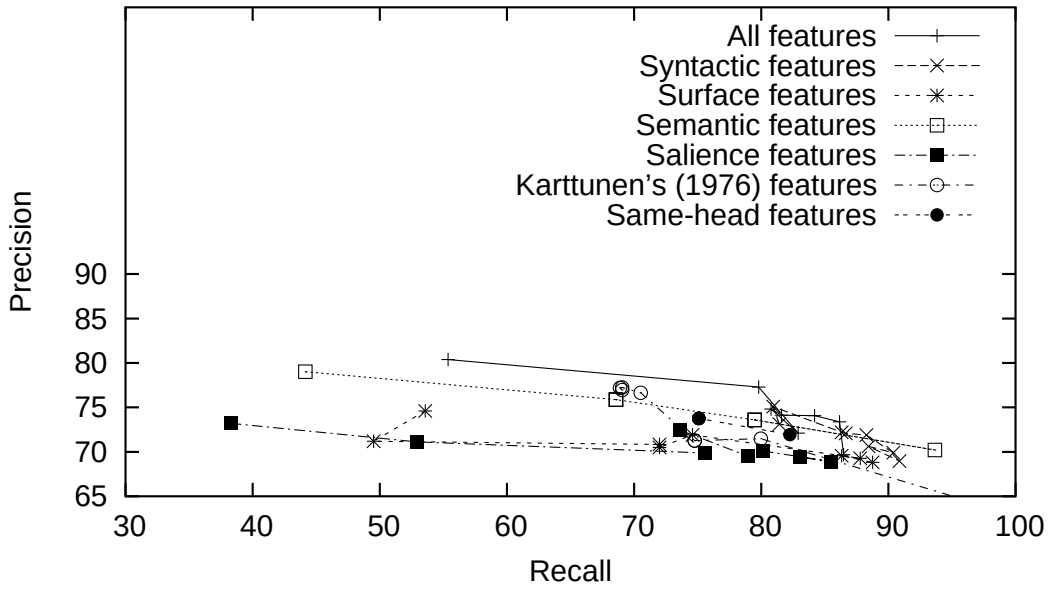
We have trained the SVM^{light}classifier for ±*discourse_new* descriptions. Its performance is summarized in Table 7.2. Compared to the baseline, the recall goes down (which is not surprising as the baseline assigns the +*discourse_new* label to all the instances and therefore has the recall level of 100%), but the precision improves significantly. This results in an F-score improvement of around 5-8%, corresponding to 23-38% relative error reduction.

Among different feature groups, surface, salience, and Karttunen's (1976) factors show virtually no performance gain over the baseline. Surface features are too shallow (recall that they encode such knowledge as "the markable is a string of digits" etc). Salience and Karttunen (1976)-motivated features have primarily been designed to account for the probability of a markable being an antecedent, not an anaphor. Based on semantic features alone, the classifier does not perform different from the baseline — although, by bringing the recall and precision values closer together, the F-score improves, the precision is still low.

The two groups with the best precision level are syntactic and "same-



(*a*)



(*b*)

Figure 7.1: Ripper-based anaphoricity (a) and antecedenthood (b) detector: performance on the validation (3 MUC-7 "train" documents) data for different values of the Loss Ratio parameter.

Features	Test data			Validation data		
	Recall	Precision	F	Recall	Precision	F
Baseline	100	66.52	79.89	100	62.81	77.16
All	††95.72	*69.23	80.35	††91.53	*68.04	78.06
Surface	††94.56	68.50	79.45	††93.49	63.78	75.83
Syntactic	††95.72	*69.23	80.35	††91.53	*68.04	78.06
Semantic	††94.92	*69.41	80.18	††93.65	**70.21	80.25
Salience	††98.88	67.00	79.88	††98.86	62.84	76.84
Same-head	100	66.52	79.89	100	62.81	77.16
Karttunen's	††99.29	67.31	80.23	††98.86	63.49	77.32

Table 7.3: An SVM-based antecedenthood detector: performance for the $-ante$ class on the validation (3 MUC-7 "train" documents) and test (20 MUC-7 "formal" documents) data.

head" features. In fact, the classifier based on these features alone (Table 7.2, last line) achieves almost the same performance level as the one based on all features taken together (no significant difference in precision and recall, χ^2-test).

As we have already mentioned when discussing the baseline, from a coreference resolution perspective, we are interested in a discourse-new detector with a high precision level: each anaphor misclassified as *discourse_new* is excluded from further processing and therefore cannot be resolved. On the contrary, if we misclassify a non-anaphoric entity as *discourse_old*, we still can hope to correctly leave it unresolved by rejecting all the candidate antecedents. Therefore we might want to improve the precision of our discourse-new detector as much as possible, even at the expense of recall.

To increase the precision level, we have chosen another machine learner, Ripper, that allows to control the precision/recall trade-off by manually optimizing the *Loss Ratio* parameter. Figure 7.1a shows Ripper's performance on the validation data for different feature groups: by varying the Loss Ratio from 0.3[3] to 1.0, we obtain different precision and recall values. As these classifiers are learned and evaluated to find optimal prefiltering settings for the main coreference resolution engine, we do not use our test data here.

As in SVM's case, the best performing groups are syntactic and "same-head" features. With all the features activated, the precision gets as high as 90% when the Loss Ratio is low. In our next experiment (Section 7.2.2) we will see if this performance is reliable enough to help a coreference resolution engine.

[3]Lower values result in the trivial labelling ("classify everything as *discourse_old*").

Identifying non-antecedents. We have trained another family of classifiers to detect non-antecedents. Table 7.3 shows SVM's performance for the $\pm ante$ task. The major class labelling, $-ante$ serves as a baseline. The classifier's performance is lower than for the $\pm discourse\_new$ task, with only syntactic and semantic features leading to a significant precision improvement over the baseline.

The lower performance level reflects the intrinsic difficulty of the task. When processing a text, the reader has to decide if an encountered description is a re-mention or a new entity to be able to correctly ground it in the discourse model. Therefore we can expect linguistic cues to signal if a markable is $\pm discourse\_new$. For $\pm ante$ descriptions, on the contrary, there is no need for such signals: often an entity is introduced but then never mentioned again as the topic changes.

As Table 7.3 shows, the classifier mostly makes precision errors. For non-antecedents, precision is not as crucial as for non-anaphors: if we erroneously discard a correct antecedent, we still can resolve subsequent anaphors to other markables from the same chain:

(93) The competition is even tougher for Aerospatiale in that [the U.S. dollar]$_1$ has weakened 10 percent against the French franc last year, giving U.S. companies what Gallois called a "superficial" advantage. <..>
"I have to say that our target is to be a profitable company," he said, "even if [the dollar]$_{2,ante=1}$ is [a handicap]$_{3,ante=2}$ for us. With [the U.S. dollar]$_{4,ante=3}$ at [five francs]$_{5,ante=4}$, we have to be profitable, at least in 1998."

Five markables form a coreference chain in this snippet: "the U.S. dollar", "the dollar", "a handicap", "the U.S. dollar", and "five francs". Even if we misclassify, for example, the second and the third markable as $-ante$, we still can correctly resolve the fourth anaphor by linking it to the first NP in the chain. However, if we misclassify the first markable and discard it from the pool of antecedents, we have no chance to correctly resolve the second anaphor. And, by discarding the second markable, we need much more sophisticated inference to correctly resolve the third anaphor.

Consequently, we would still prefer recall errors over precision errors, although not to such extent as for the $\pm discourse\_new$ classifier. We have trained a family of Ripper classifiers to improve the precision level by decreasing the Loss Ratio parameter from 1.0 to 0.3. Their performance on the validation data is shown on Figure 7.1b. The best observed precision level is 80.4% for the "all features" classifier.

For the same reasons as above, we do not evaluate the classifiers on the test data.

To summarize, the present experiment shows that automatically induced

classifiers, both SVM and Ripper-based, can successfully identify unlikely anaphors and antecedents. The performance level (F-score) varies around 75 − 88% for different test sets (validation vs. testing) and tasks ($\pm$*discourse_new* vs. $\pm$*ante*).

7.2.2 Experiment 10: Integrating Anaphoricity and Antecedenthood Prefiltering into a Coreference Resolution Engine

In the previous experiment we have learned two families of classifiers, detecting unlikely anaphors and antecedents. In this section we incorporate our classifiers into a baseline coreference resolution system. Throughout this thesis, we use an SVM classifier with Soon et al.'s (2001) features as a baseline.

Oracle settings. To investigate the relevance of anaphoricity and antecedenthood for coreference resolution, we start by incorporating oracle-based prefiltering into the baseline system. For example, our oracle-based anaphoricity filter discards all the discourse-new markables (according to the MUC-7 coreference chains) from the pool of anaphors.

The impact of our ideal filters on the main system is summarized in Table 7.4. As expected, by constraining the set of possible anaphors and/or antecedents, we dramatically improve the algorithm's precision. Slightly unexpected, the recall goes down even in the oracle setting. This reflects a peculiarity of the MUC-7 scoring scheme — it strongly favors long chains. Prefiltering modules, on the contrary, split long chains into smaller ones. Consider the following example:

(94) COMPANY SPOTLIGHT: MCDONALD'S SHOPS FOR CUSTOMERS AT [WAL-MART]$_1$

> Todd Purvis couldn't see through the sleet and grime on his windshield, so he stopped at [a Wal-Mart]$_2$ in North Brunswick, New Jersey, to buy washer fluid. Inside, he saw those golden arches. In minutes, Purvis was sitting in a McDonald's wolfing down a cheeseburger and chugging a Coke. < .. >
> McDonald's Corp. is shopping for customers inside some of the nation's biggest retailers, including [Wal-Mart Stores Inc.]$_{3,ante=1}$ and Home Depot Inc.

Our baseline system erroneously merges three descriptions, "WAL-MART"$_1$, "a Wal-Mart"$_2$, and "Wal-Mart Stores Inc."$_3$ into one chain by resolving the second markable to the first and the third to the second. The correct partition in this case would result in two chains — for the Wal-Mart company (M_1 and M_3), and for a particular Wal-Mart store (M_2). If we now turn the anaphoricity

Prefiltering	Recall	Precision	F-score
No prefiltering (baseline)	54.5	56.9	55.7
Ideal *discourse_new* detector	49.6	**73.6	59.3
Ideal *ante* detector	54.2	**69.4	60.9
Ideal *discourse_new* and *ante* detectors	52.9	**81.9	64.3

Table 7.4: Incorporating oracle-based ±*discourse_new* and ±*ante* prefiltering into a baseline coreference resolution system (an SVM classifier with Soon et al.'s (2001) features: performance on the validation data (3 MUC-7 "training" documents).

filter on, the system proposes two chains, {WAL-MART} and {a Wal-Mart, Wal-Mart Stores Inc}: the second markable is +*discourse_new* and shouldn't be resolved. As a result, two anaphoric markables, M_1 and M_3, belong to different chains in this new partition and the recall level goes down.

Several other studies (Ng and Cardie, 2002b; Mitkov et al., 2002) have revealed similar problems: existing coreference scoring schemes can not capture the performance of an anaphoricity classifier. Bagga and Baldwin (1998a) also argue that the MUC scoring scheme fails to account for "non-coreferent" relations, thus, favoring systems with higher recall and lower precision levels.

With precision getting much higher at the cost of a slight recall loss, the ideal ±*discourse_new* and ±*ante* detectors improve the baseline coreference engine's performance by up to 10% (F-score).

Automatically acquired detectors. Getting from the oracle setting to a more realistic scenario, we have combined our baseline system (an SVM classifier with Soon et al.'s (2001) features) with the ±*discourse_new* and ±*ante* detectors we have learned in Experiment 9.

The evaluation has been organized as follows. For a given Loss Ratio value, we have learned a ±*discourse_new*/±*ante* detector as described in Experiment 9 above. The detector is then incorporated as a pre-filtering module into the baseline system. This allows us to evaluate the performance level of the main coreference resolution engine (the MUC score) depending on the precision/recall trade-off of the pre-filtering modules.

The results (Figure 7.2) show that automatically induced detectors drastically decrease the main system's recall: it goes down to 40% (for ±*discourse_new*, $L = 0.8$) or even 33% (for ±*ante*, $L = 1$). For small L values, the system's recall is slightly lower, and the precision higher than the baseline (both differences are not significant). The resulting F-score for the system with prefiltering is slightly lower than the baseline's performance for small values of the Loss Ratio parameter and then decreases rapidly for $L > 0.5$.

To summarize, the results of the present experiment are ambivalent. On

the one hand, ideal detectors bring F-score gains by significantly increasing the system's precision. On the other hand, error-prone automatically induced detectors are not reliable enough to produce a similar precision gain and the system's F-score goes down because of the recall loss, as the baseline's recall is already relatively low. Consequently, a coreference resolution algorithm might profit from an automatic $\pm$*discourse_new* or $\pm$*ante* detector if its precision has to be improved, for example, if it mainly makes recall errors or, for a specific application, if a high-precision coreference resolution algorithm is required (as, for example, the CogNIAC system proposed by Baldwin (1996)).

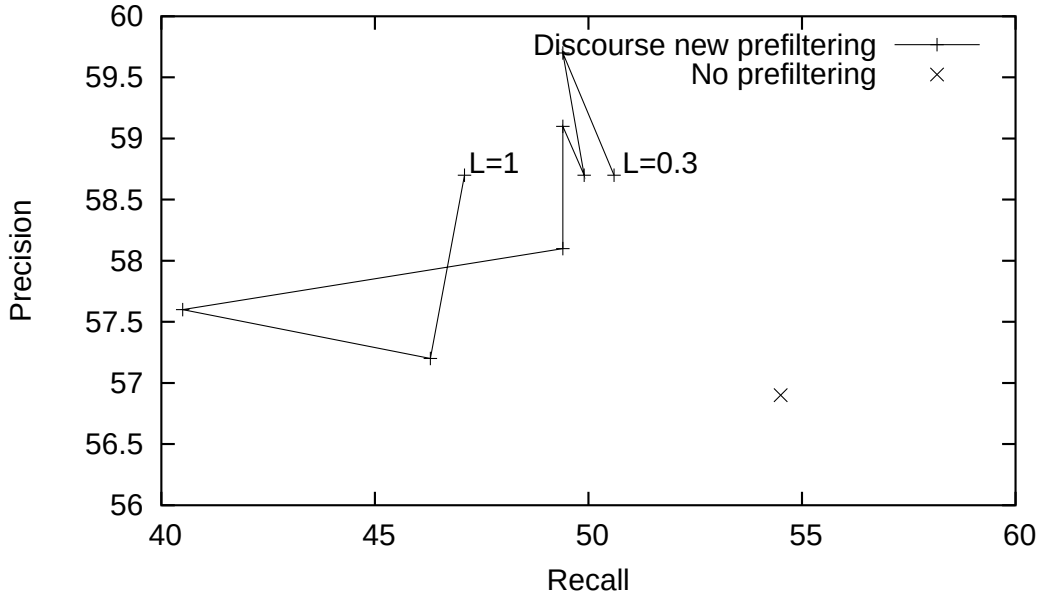
7.3 Summary

In this Chapter we have investigated the possibility of automatically identifying unlikely anaphors and antecedents. As only around 30% of markables in newswire texts participate in coreference chains, our $\pm$*discourse_new* and $\pm$*ante* detectors might significantly constrain the main algorithm's search space, improving its speed and performance.

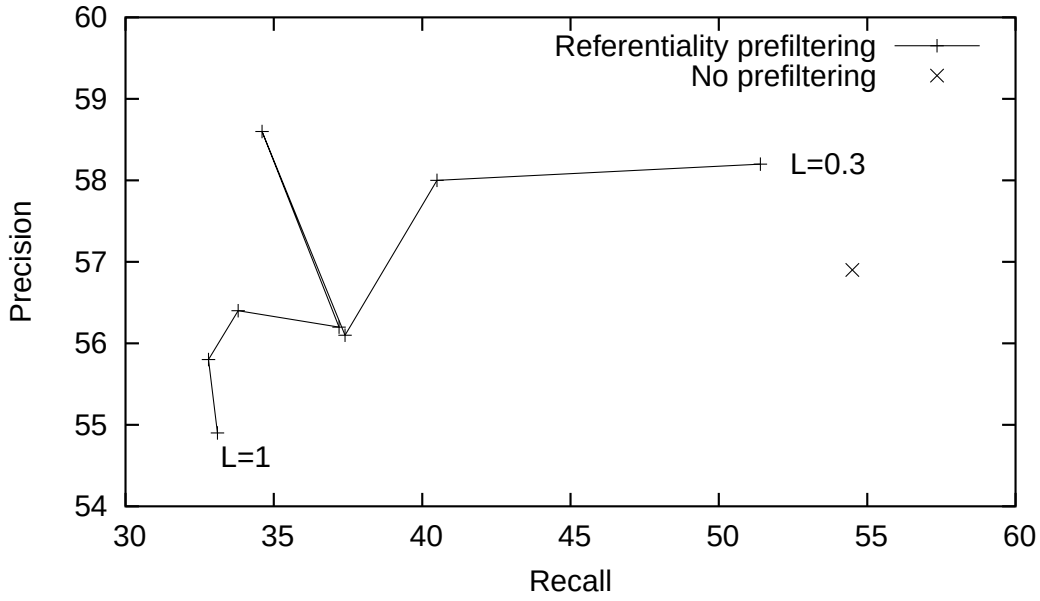
In Experiment 9 (Section 7.2.1), we have compared different feature groups for the tasks of $\pm$*discourse_new* and $\pm$*ante* detection. We have seen that, for both tasks, SVM and Ripper classifiers based on all the investigated features outperform the baseline. We have also learned two families of classifiers with different precision/recall trade-offs.

In Experiment 10 (Section 7.2.2), we have incorporated our $\pm$*discourse_new* and $\pm$*ante* detectors into a baseline coreference resolution system. We have seen that ideal prefiltering significantly improves the system's precision at the expense of a slight recall loss. This leads to an F-score improvement of up to 10%. Automatically acquired detectors can only moderately improve the system's precision and therefore do not bring any F-score gains.

We still believe, however, that anaphoricity and antecedenthood detectors might help a coreference resolution system with a lower precision and higher recall.



(a)



(b)

Figure 7.2: A baseline coreference resolution engine augmented with Ripper-based anaphoricity (a) and antecedenthood (b) prefiltering: performance on the validation (3 MUC-7 "train" documents) data for different Loss Ratio (L) values of prefiltering classifiers.

Chapter 8

Combining Different Knowledge Types

In the previous chapters we have investigated the interaction of different linguistic factors with coreference. We have seen that various linguistic parameters affect the distribution of anaphoric links in a document, but neither name-matching, nor syntactic, nor semantic, nor salience features alone are sufficient to build a coreference resolution system. Each feature group, however, encodes valuable information that may help improve the performance level of a baseline algorithm (Soon et al., 2001). This is in accordance with theoretical studies claiming that coreference is a complex phenomenon involving different kinds of linguistic knowledge. In the present chapter we combine all the investigated factors to build a linguistically informed coreference resolution system.

Different types of anaphoric links may require very specific processing:

(95) Washington, April 4 (Bloomberg) – The Navy is considering a new ship that would carry on the shore-pounding tradition of World War II's famous battleships. Although battleships were consigned to history, the need to attack an enemy's positions from the sea while soldiers attack over land "is there," said Chief of Naval Operations Adm. J.M. Boorda during a meeting with reporters in his Pentagon office...
Shipbuilders may also take a back seat to the weapons makers and the systems integration companies because Boorda wants to limit the total number of sailors on the arsenal ship to between 50 and 60. "A big part of the savings from this ship is going to be that it's manned with fewer people," the admiral said. "People are the most important thing but they're also the most expensive life cycle part of the whole thing. We want to do with technology what we can to keep the numbers of people down."

This example illustrates the variety of coreference links in our data. Some markables have a similar surface form ("Adm. J.M. Boorda" and "Boorda"; "a new ship", "the arsenal ship", and "this ship"), some participate in specific syntactic constructions ("people" and "the most important thing"), some are semantically very close ("a new ship" and "the battleships"). Salience factors are important for pronominal anaphors ("they", "it", "we", etc). We have to take into account a variety of linguistic parameters to build a full-scale coreference resolution engine.

The system we use a a baseline, a reimplementation of (Soon et al., 2001), relies on just 12 simple features. It can therefore resolve easy anaphors in (95), for example ("Adm. J.M. Boorda", "Boorda"), but more difficult links remain problematic. Difficult anaphors are mostly left unresolved: the baseline, for example, has no features to account for predicate nominals ("the most important thing") or pronouns ("they", "we") and suggests no antecedents. The baseline also occasionally produces spurious links: it resolves "April 4" to "Washington", considering these markables to be parts of an apposition. We have to rely on more sophisticated features to resolve difficult anaphors – such new features should encode linguistic factors not covered by the baseline and provide more accurate representation for the information already in use.

Although a learning-based system could benefit from a richer feature set, two following problems may potentially arise:

1. Robustness and noise. More sophisticated features cannot be extracted reliably. Automatic extraction of sophisticated features inevitably leads to a noisy dataset.

2. Overfitting. A large feature set may result in a too detailed representation of the training data, if the machine learner has a poor build-in control for capacity (see Section 2.4).

We investigate the usability of our rich feature set in empirical evaluation (Experiment 11, Section 8.1). To allow fair comparison, we run several machine learners on a standard corpus (MUC-7) with a traditional set-up (the setting used by Soon et al. (2001)). We also provide a detailed error analysis (Section 8.2) and discuss the observed problems and possible directions for future work (Section 8.3).

8.1 Experiment 11: Coreference Resolution with and without linguistic knowledge

In the previous chapters we have seen how various linguistic properties of noun phrases interact with coreference. We have investigated surface, syntactic, semantic, and pragmatic factors relevant for our problem. In Experiments

2-8 we have built coreference resolution engines based on each feature group alone. They have yielded a moderate performance, showing that coreference resolution is a complex task requiring various kinds of linguistic knowledge.

In the present experiment we combine all investigated properties of NPs and NP pairs to build a rich linguistically-motivated feature set. Our system has 351 nominal features (1096 boolean/continuous), representing surface (122 features, see Table 3.8), syntactic (64, Table 4.15), semantic (29, Table 5.8), and salience-based (136, Table 6.11) properties of markables and markable pairs.

We use five publicly available machine learners in our experiments to be sure that the effect is not accidental. Each learner has advantages and disadvantages for our task.

RIPPER (Cohen, 1995) is an information gain-based rule induction system. The main advantage of Ripper for coreference resolution is that its rules can be composed of only very few features. It allows RIPPER to capture coreference links signaled by a single feature (for example, two parts of the copula construction are coreferent, even if they seem to have incompatible properties) The main disadvantage of Ripper is that it is very unstable: even a minor change in the feature set can potentially result in a major rearrangement of the learned classifier, and, thus, affect the system's performance in a rather unpredictable way.

SLIPPER is a newer, improved algorithm based on RIPPER and confidence rate boosting.

C4.5 (Quinlan, 1993) is a decision tree learner. For our task it has essentially the same advantages and disadvantages as RIPPER. C4.5 is not very effective when some features (for example, grammatical roles) have a lot of not uniformly distributed nominal values. Most state-of-the-art learning-based Coreference Resolution algorithms (McCarthy and Lehnert, 1995; Vieira, 1999; Soon et al., 2001) rely on decision trees.

SVM^{light} (Joachims, 1999) is an implementation of Support Vector Machines (Vapnik, 1995), that are known for their high performance, especially for NLP tasks. In particular, SVMs have a built-in capacity control to deal with overfitting. This is especially important for our extended feature set.

Maxent (Le, 2004) is an implementation of GIS Maximum Entropy modeling (Curran and Clark, 2003a). Similar to SVMs, ME-based classifiers, being the most non-committal models, are less prone to overfitting.

We learn classifiers for two different feature sets to build coreference resolution engines with and without linguistic knowledge, as described below.

8.1.1 Baselines

We compare our linguistically motivated approach to a set of naive baselines and a more intelligent baseline (Soon et al., 2001).

Naive Baselines. Our first naive baseline is the "merge all" algorithm: all markables in a document are linked together to form a single coreference chain. Its performance is shown in Table 8.1 (first row). The "merge all" strategy yields a relatively high F-score (two of the seven MUC-7 systems have worse performance figures). However, it is obviously not an acceptable solution for the coreference resolution task, as the resulting partition, i.e. one chain, is not informative. This suggests that the F-score measure might be not the best evaluation criterion for coreference resolution systems[1].

The baseline's recall is remarkably low: one would expect a recall level of 100% when *all* markables in a document form a single coreference chain. The lower value (85%) reflects numerous discrepancies between our automatically extracted markables and those suggested by the MUC-7 annotators. We will come back to this point in Section 8.2.

Table 8.1 also shows performance figures for several one-feature baselines[2]. Only the matching features (surface or head matching) achieve a reasonable performance level. Salience features lead to random partitioning, as the example of `sfirst_prev_agree`[3] clearly shows. Semantic one-feature baselines link almost all markables together, performing similar to the "merge all" algorithm. We have also observed the same pattern for syntactic agreement-based one-feature classifiers. Explicit syntactic indicators of coreference (apposition and copula) help us create classifiers with an acceptable precision, but a very low recall level: this is not surprising, because such constructions cover only a small proportion of coreference links.

Intelligent Baseline: Reimplementation of (Soon et al., 2001). Soon et al. (2001) have presented the first full-scale learning-based coreference resolution system, achieving a performance level comparable or even outperforming the best (knowledge-based) systems in the MUC-7 competition. It relies on just 12 simple features, shown in Table 8.2. The algorithm of Soon et al. (2001) is described in detail in Chapter 2 and briefly summarized below.

The coreference classifier is created as follows. Each anaphor in the training corpus is paired with its closest antecedent to create a positive instance and with all the markables in between to create negative instances. The feature vectors for these instances are submitted to the C5.0 decision tree learner. The C5.0 internal parameters (pruning level and the minimum number of instances per leaf node) are optimized with 10-fold cross-validation.

The C5.0 learner outputs a decision tree that is applied to the test corpus. For each candidate anaphor in the corpus, test instances are constructed by pairing it with the preceding markables (starting with the closest one and

[1] See (Bagga and Baldwin, 1998a), Section 2.2 and Chapter 7 for related discussions.

[2] Cf. Tables 3.8, 4.15, 5.8, and 6.11 for the features' descriptions.

[3] The antecedent is the first markable in a sentence, the anaphor is some markable in the next sentence; they have compatible agreement values.

	Test set		
Features	Recall	Precision	F-score
"merge all"	85.0	33.6	48.2
One-feature baselines			
same_surface	31.7	72.9	44.2
same_head	44.1	58.3	50.2
same_surface_normalized	42.3	73.5	53.7
same_head_normalized	52.4	59.0	55.5
apposition	2.3	69.2	5.4
copula	2.0	65.0	3.9
syntagree	81.2	33.0	46.9
sfirst_prev_agree	25.6	22.2	23.8
compatible_semclass	74.1	31.9	44.6

Table 8.1: Naive baselines: performance on the test data (20 MUC-7 "formal test" documents").

Feature	Values	Description
DIST	0..n	distance in sentences between *ana* and *ante*
I_PRONOUN	0,1	*ante* is a pronoun
J_PRONOUN	0,1	*ana* is a pronoun
STR_MATCH	0,1	*ana* matches *ante* (stripping off determiners)
DEF_NP	0,1	*ana*'s determiner is "the"
DEM_NP	0,1	*ana*'s determ. is "this," "that," "these,", "those"
NUMBER	0,1	*ana* and *ante* agree in number
SEMCLASS	0,1,?	*ana* and *ante* have compatible semantic classes
GENDER	0,1,?	*ana* and *ante* agree in gender
PROP_NAME	0,1	*ana* and *ante* are both proper names
ALIAS	0,1	*ana* is an alias of *ante* or vice versa
APPOSITIVE	0,1	*ana* is in apposition to *ante*

Table 8.2: Features used by Soon et al. (2001)

Learner	(Soon et al., 2001) features			Our features		
	R	P	F	R	P	F
Ripper	44.6	**74.8	55.9	**65.8	51.1	57.5
Slipper	84.7	33.8	48.4	85.8	33.9	48.6
C4.5	53.5	**72.8	61.7	**65.1	64.1	64.6
SVM^{light}	50.9	68.8	58.5	**63.9	67.0	65.4
Maxent	49.2	64.1	55.7	50.5	**72.2	59.4
"Merge all" baseline	85.8	33.9	48.6	85.8	33.9	48.6
(Soon et al., 2001) system	56.1	65.5	60.4	N/A	N/A	N/A

Table 8.3: Performance on the test data (20 MUC-7 "formal test" documents) for different feature sets, training on all the training instances. Significantly better recall and precision figures are marked by ** (χ^2-test, $p < 0.01$) for each machine learner correspondingly.

proceeding backwards). The test instances are submitted to the classifier. Once an instance is classified as positive, it is annotated as the antecedent for the anaphor in question, and the algorithm goes on to the next candidate anaphor.

We use the same feature set and the same setting in our reimplementation. However, we train not only a decision tree-based classifier, but also several others. Experimentation on optimizing numerous learning parameters lies outside the scope of this thesis. Our results are slightly different from those reported in Soon et al. (2001) even for decision trees, as we have different procedures for extracting markables from raw texts and for computing feature values. For example, we can more or less straightforwardly reimplement Soon et al.'s (2001) feature "*ana* is a pronoun", but not "*ana* and *ante* are both proper names" – the latter relies on the chosen NE-tagger.

8.1.2 Performance and Learning Curves

Table 8.3 shows the system's performance for our two different feature sets: the one advocated by Soon et al. (2001) and the extended set containing all the features discussed in the thesis (Tables 3.8, 4.15, 5.8, and 6.11).

The F-scores range from 55% to 65% (with the exception of SLIPPER, see below). It must be, however, noted that the upperbound for the MUC-7 coreference resolution task lies far below 100%. We have seen, for example, that the "merge all" strategy yields a recall value of 85%, although it should, theoretically, resolve *all* anaphors. Inconsistencies of the MUC annotations and our parsing and tagging modules do not allow us even to attempt resolution for the remaining 15% of anaphors. This will be described in details in the

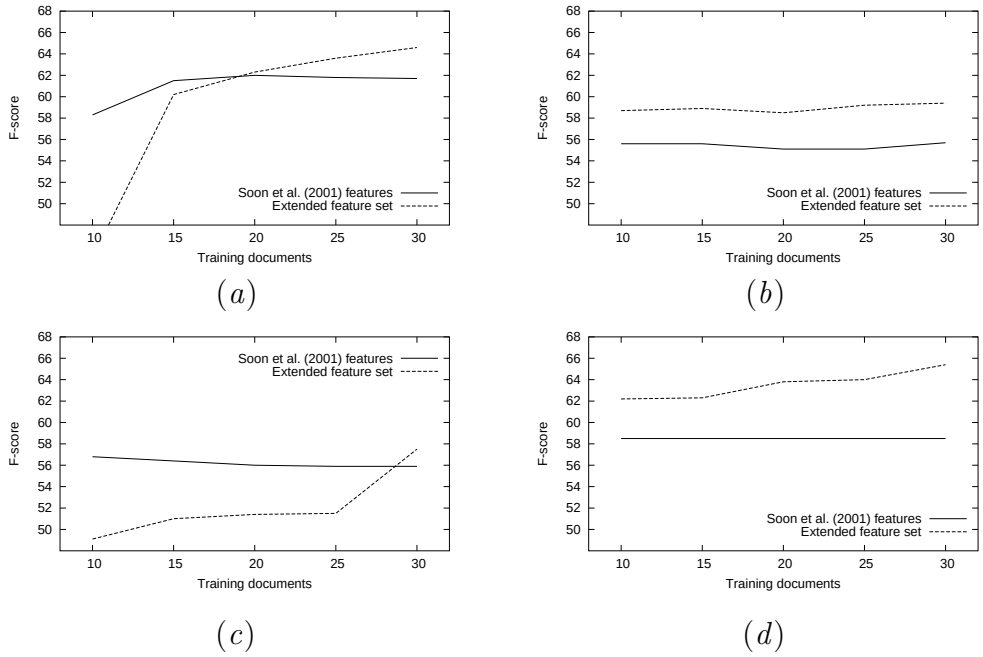


Figure 8.1: Learning curves (F-score) for different machine learning algorithms on the MUC-7 data: (a) C4.5, (b) Maxent, (c) Ripper, (d) SVM^{light}.

following section.

The SLIPPER learner could not resolve the problem: for both feature sets, the SLIPPER classifier merges virtually all the markables into one chain, and, thus, performs at the naive baseline's level. All the other learners show better performance when a richer feature set is used. The most substantial improvement is achieved by support vector machines[4]. For all the learners, the recall goes up reflecting the system's ability to resolve more difficult anaphors (significant for Ripper, C4.5, and SVM). The precision, however, goes down indicating that such difficult anaphors can be resolved only with substantial noise (significant for Ripper, C4.5).

A remarkable exception is the Maximum Entropy classifier: both its recall and precision go up when we add linguistically motivated features. However, the recall goes up only slightly, indicating that the system has not acquired new cases of anaphora, but, instead, has learned a more accurate classification for the old ones.

We have conducted a second series of experiments to investigate why our

[4]To our knowledge, the system's performance with the extended feature set and the SVM^{light}learner (F-score of 65.4%) is the best up-to-date result on the MUC-7 data reported in the literature.

extended feature set leads only to a moderate improvement. Our hypothesis is that the training corpus is too small to learn more sophisticated patterns and the feature set extension leads to the overfitting problem that hinders the performance.

We have learned classifiers from the first 10, 15, 20, 25, or all the 30 "dry-run" documents to see how the system's performance depends on the amount of training data. The resulting learning curves (F-score) are shown on Figure 8.1.

The curves clearly suggest that even very few documents are sufficient to learn a reliable classifier with the Soon et al.'s (2001) features. The performance remains on essentially the same level or sometimes even goes down (Ripper, MaxEnt) when we increase the amount of training data. For example, the SVM learner has created exactly the same classifier with just the first 10 vs. all the 30 training documents. The learners have evidently extracted all the information encoded in the basic 12 features and have no additional knowledge to improve any further.

For the extended feature set, on the contrary, the performance is very low when only 10 training documents are available, indicating a high level of overfitting. With more training material, the extended feature set leads to better and better classifiers, showing no sign of convergence. This suggests that one can get a better coreference resolution algorithm using our linguistically motivated feature set by annotating more documents.

Table 8.4 compares the performance of our system with the extended feature set to single-source algorithms, relying on name-matching, syntactic, semantic or salience knowledge exclusively. The SVM^{light}module is used for machine learning. The results show that a combination of different knowledge sources significantly outperforms each individual single-source classifier. This is in accordance with theoretical research claiming that coreference is a complex phenomenon affected by numerous linguistic factors.

Our system with the extended feature set and the SVM^{light} learning module achieves the best performance (F-score of 65.4%) on the MUC data, reported in the literature, outperforming other statistical and rule based approaches, evaluated on this dataset, as well as single-source classifiers.

8.2 Error Analysis

Our coreference resolution engine achieves the best performance, reported for the MUC-7 corpus in the literature. But still there is room for improvement. We have identified a number of problematic cases where our algorithm systematically suggests spurious antecedents or misses the correct ones. They can be roughly divided into three major classes: inconsistencies of the annotated material, errors aggregated from the preprocessing modules and deficiencies of the

	Test set		
Features	Recall	Precision	F
all	63.9	67.0	65.4
matching	††52.2	††61.2	56.3
syntax	††7.6	68.5	13.8
semantics	††28.5	††48.3	35.9
salience	**86.6	††35.2	50.0

Table 8.4: Performance on the test data (20 MUC-7 "formal test" documents) for different feature sets, SVM^{light}, training on the 30 MUC-7 "dry-run" documents. Significant improvement and deterioration of recall/precision figures (compared to the whole extended feature set) are marked by ** and †† (χ^2-test, $p < 0.01$).

coreference resolver proper (either of its learning component or of the whole framework).

In this section we focus on a detailed error analysis for our best classifier – the SVM^{light} learner with the extended feature set. We distinguish between several types of errors, mainly following linguistic criteria. The error groups and the corresponding figures are presented in Tables 8.5 and 8.7 below.

In the next section we will rely on this detailed analysis to discuss the three major error classes and suggest possible remedies on a more general level.

8.2.1 Recall Errors

A recall error occurs when a system fails to produce a link suggested by the manually annotated data. The following problems may potentially lead to a missing link:

1. The (manually annotated) anaphor is not recognized by our preprocessing modules (i.e. it does not belong to the set of automatically extracted markables, see Section 2.5).

2. None of the preceding manually annotated markables in the anaphor's chain is recognized by our preprocessing modules. Note that even if the exact MUC-7 antecedent does not belong to our markables pool, the system can still correctly resolve the anaphor to some earlier antecedent.

3. The systems fails to link two correctly extracted markables.

Below we focus on the recall errors made by our SVM^{light}-based system on the MUC-7 data (20 "formal" documents). In our examples of recall errors, we show the original MUC-7 SGML annotation. We have removed all the

irrelevant mark-up for simplicity, leaving only the markable under discussion and its (manually annotated) coreference chain.

We do not expect our system to produce an exact copy of the annotated data — the scorer operates on whole chains instead of links. When our system fails to link an anaphor to *any* of the possible antecedents, we choose only the intuitively easiest (requiring less reasoning) link to analyze as a recall error.

MUC-7 Inconsistencies. We have found a few very questionable markables and links in the MUC-7 test corpus. Such cases can not be covered by our system and thus decrease its performance (see below for MUC-related precision errors). We have also found a specific mark-up pattern that is analyzed incorrectly by the MUC-7 scoring program, leading to a systematic performance loss.

We have encountered several inconsistencies in marking up pre- and post-modified noun phrases. The MUC-7 annotation guidelines suggest that "the text element to be enclosed in SGML tags is the maximal noun phrase; the head will be designated by the MIN attribute". Some markables, however, are annotated too short:

(96) "If some countries try to block China <COREF ID=71 REF=72 MIN= "accession">WTO accession</COREF> on the basis of the military exercises, that will not be popular and will fail to win the support of other countries," she said.

(97) Guitarist Courtney Taylor and synthist <COREF ID=55> Zia McCabe </COREF> meshed beautifully – complex harmonics shimmering off simple chords ...

Our system has suggested correct markables, "China WTO accession" and "synthist Zia McCabe" here, but they can not be aligned with the annotated constituent ("WTO accession", "Zia McCabe") by any automatic scoring program.

Some markables do not have a (necessary) MIN argument:

(98) Not when it comes to <COREF ID=0 REF=1> Love&Rockets, <COREF ID=2 REF=0 MIN="trio"> the British trio that was <COREF ID=3 REF=2 MIN="offshoot"> an offshoot of the early '80s goth band Bauhaus </COREF> and was last glimpsed on the charts in 1989 </COREF> </COREF>.

A coreference resolution system might correctly resolve "the British trio" to "Love&Rockets" in (98) but the scorer would still consider it an error, because the antecedent, according to the manual annotation, should be the maximal

	Errors	%
MUC-7 inconsistencies	17	3.6%
Misleading markables	166	35.4%
auxiliary doc parts	50	10.7%
tokenization	8	1.7%
one-word modifiers	36	7.7%
multi-word modifiers	10	2.1%
bracketing/labelling	54	11.5%
other	8	1.7%
Propagated P-errors	31	6.6%
PRO-anaphora	17	3.6%
NP-anaphora	14	3.0%
Pronominal anaphora	77	16.4%
NE-matching	31	6.6%
Syntactic constructions	39	8.3%
apposition	18	3.8%
copula	8	1.7%
quantitative	13	2.8%
NP-anaphora	104	22.2%
same head	4	0.9%
morph. variants	7	1.5%
head-modifier	10	2.1%
NP_{ana}-NP_{ante}	46	9.8%
NP_{ana}-NE_{ante}	28	6.0%
NE_{ana}-NP_{ante}	7	1.5%
NE_{ana}-NE_{ante}	6	1.3%
total	469	100%

Table 8.5: Performance of the SVM^{light}classifier: recall errors on the testing data (20 MUC-7 "formal test" documents).

NP, "Love&Rockets, the British trio that was an offshoot of the early '80s goth band Bauhaus and was last glimpsed on the charts in 1989."

A similar problem occurs in Example (99), where the MIN value for "Jim Johannesen, vice president of site development for McDonald's" is not correct:

(99) ... said <COREF ID=56 MIN="vice president">Jim Johannesen, <COREF ID=55 REF=56 MIN="vice president"> vice president of site development for McDonald's </COREF></COREF>.

Strictly speaking, a coreference resolution engine could output longer markables, not relying on the MIN argument for scoring. This solution, however, is very sensible to parsing errors that are more likely to occur with larger constituents. It is also not plausible from an Information Extraction perspective: for Examples (98, 99), our system has suggested correct anaphoric links, bringing some new information about named entities: for example, we get to know that "Jim Johannesen" is a "vice president of site development for McDonald's" and can add this fact to the other knowledge on "Jim Johanessen" our system has collected so far. Such information can be further processed by IE applications. It is therefore desirable to suggest NEs (and not embedding constituents) as markables.

The following example illustrates a problem with the MUC scoring module. Two embedded markables, "the next generation of video games with 3-D images" and "video games" have the same MIN value – "games". It makes the scorer completely ignore the internal markable: our system has correctly proposed "video games" as an anaphor and correctly found its antecedent. The scorer, however, has aligned our "video games" constituent with the embedding NP, "the next generation of video games with 3-D images", and reported two errors: incorrectly resolved sixth markable (precision error) and not resolved eighth markable (recall error):

(100) On Monday, industry sources said, Mountain View-based Silicon Graphics Inc. will release a technology dubbed "FireWalker" designed to make <COREF ID=6 REF=7 MIN="games">the next generation of <COREF ID=8 REF=9 MIN="games">video games</COREF> with 3-D images</COREF> more economical and commonplace.

Finally, spurious coreference links have sometimes been suggested by the annotators. We have found only very few clearly erroneous examples. One of them is shown below – the non-referential pronoun "it" is incorrectly linked to the chain for "new retailers":

(101) Keep an eye out for <COREF ID=161 MIN="retailers"> hot new retailers </COREF> making <COREF ID=160 REF=161> their

</COREF> way to the top – don't be surprised if there's a McDonald's inside when <COREF ID=163 REF=160>it</COREF> gets there.

We have also observed a number of questionable links, see, for example, (104) and the following discussion. In case of any doubt we have always considered the manual annotation to be accurate and suggested another error type.

To summarize, around 4% of our recall errors result from inconsistencies in the annotated test material. They include mainly incorrect mark-up for individual markables (bracketing and MIN values), but also a few spurious links.

Errors in Markables' Determination We rely on (automatic) external modules for computing our markables' set (see Section 2.5). This means that at least some MUC-7 markables are inevitably missed by our system. Such markables never get submitted to the classifier and therefore decrease the system's recall. Typical problems in automatic markable extraction are discussed below.

Each MUC-7 document contains auxiliary parts (TEXT, SLUG, DATE, NWORDS, PREAMBLE, and TRAILER should be annotated, see Section 6.2). These segments have their own structure that cannot be parsed reliably. We treat hyphenation marks as word boundaries (except for DATE) and then consider each word in SLUG, DATE and NWORDS to be a markable and submit PREAMBLE to the parser. This naive solution is not always accurate:

(102) <SLUG fv=taf-z> BC-<COREF ID=1>LORAL-SPACE</COREF>-470&ADD-N </SLUG>

Our system has suggested two markables, "LORAL" and "SPACE" instead of "LORAL-SPACE". When "Loral Space" is re-mentioned further in the document, the classifier cannot find any suitable antecedent for it and leaves it unresolved. Extracting entities from semi-structured documents is a non-trivial task of its own. It is a very data-specific problem and therefore we do not address it in this thesis.

Markables in the main document body are collected by merging the output of Charniak's (2000) parser (noun phrases and possessive pronouns) and Curran and Clark's (2003b) NE-tagger (named entities). Some MUC-7 units are either never suggested by such modules (see Examples 103 and 104) or not selected to the markables' pool by our system (105, 106). Other MUC-7 markables can potentially be recognized by an ideal parser or tagger, but the modules we use fail to analyze them (107-109)

The MUC-7 annotation guidelines imply a too fine-grained tokenization scheme – some markables are not separate words, but rather parts of compounds:

(103) They said they will start marketing the package in mid-<COREF ID=21> August</COREF> and the equipment needed to receive it will go on sale at the end of <COREF ID=20 REF=21 MIN="month"> that month</COREF>.

The markable "August" here cannot be recognized by state-of-the-art parsers, as they do not segment words into smaller units. We could implement a mini-tokenizer for splitting compounds. The data show however that the corresponding coreference links are in most cases very questionable and probably should not be annotated at all. Compare (103) to the following example:

(104) The penetration was caused either by <COREF ID=27 MIN="contaminants"> metal and dirt contaminants </COREF> or a break of one or more of the copper wires, investigators concluded in their 358-page report. ...
The <COREF ID=26 REF=27>contaminants</COREF> were either braided into the cord when it was manufactured in the mid-1980s or existed undetected within the intricate system of pulleys, electrical motors and a large spool used to store and unwind the satellite from Columbia's payload bay, the panel concluded. ...
The board found that the tether met all the mission's specifications for strength, temperature and electrical performance but noted that there was no requirement for the tether to be manufactured in a <COREF ID=59 REF=26>contaminant</COREF>-free environment.

"August" in "mid-August" (103) refers to a specific period of time and is at least in some sense coreferent with "that month". "Contaminant" in "contaminant-free" (104), on the contrary, does not refer to any discourse entity and should ideally be not linked to the descriptions of the particular contaminants discussed in the document (the 25th and 26th markables). It must also be noted that the MUC-7 annotators suggest no antecedents for non-referential full noun phrases.

Some MUC-7 markables are not full NPs, but prenominal modifiers:

(105) "Our goal is to dominate both the commercial and government parts of <COREF ID=9 REF=6>space</COREF> exploration," Walker said.

According to the MUC-7 annotation guidelines, "prenominal modifiers are markable only if either the prenominal modifier is coreferential with a named entity or to the syntactic head of a maximal noun phrase. That is, there must be one element in the coreference chain that is a head or a name, not a modifier". Most full-scale coreference resolution algorithms (Soon et al. 2001; Ng and Cardie 2002c, among others)) treat NP modifiers as markables.

Our pool of markables gets around 15% bigger when we incorporate one-word modifiers that are not parts of named entities. Even such extended candidate set cannot help us resolve some modifiers:

(106) A.The technology allows <COREF ID=56 REF=47 MIN="game">video game</COREF> makers to automatically position characters in scenes without the need to program, saving time and money.
B. On Monday, industry sources said, Mountain View-based Silicon Graphics Inc. will release a technology dubbed "FireWalker" designed to make the next generation of <COREF ID=8 REF=9 MIN="games">video games</COREF> with 3-D images more economical and commonplace... SGI's new <COREF ID=38 REF=8 MIN="game">video-game</COREF> making technology is akin to the high-tech breakthroughs that the computer workstation maker brought to moviemaking in the late '80s.

The snippet (106A) contains a two-word modifier, "video game". A coreference resolution system cannot account for such markables unless it covers all the possible combinations of adjacent one-word modifiers. This will inevitably blow up the markables' pool, deteriorating the system's precision and efficiency. The anaphor in (106B) can potentially get resolved by incorporating one-word modifiers. But if we consider "video-game" to be a one-word unit, we can hardly link it to the correct (two-word) antecedent, "video games."

We have decided not to include any NP modifiers into our markables pool, unless they are named entities. This helps us restrict the set of candidate anaphors/antecedents and therefore improve the processing time and the precision level of our system, but, in the same time, leads to some recall loss.

Some markables cannot be extracted, because the parser or the NE-tagger suggests an incorrect bracketing or labelling for the data. For example, "DreamWorks SKG" in (107) has been split by the NE-tagger into two parts – "Dreamworks" (no NE tag) and "SKG" (ORGANIZATION):

(107) Future customers may include <COREF ID=44 MIN="DreamWorks SKG">DreamWorks SKG, the Hollywood studio that plans to create movies, video games and other interactive entertainment</COREF>.

"Mesmerizing set" in (108) is not a markable, because the parser considered "set" to be a verb (labelled VBD – verb, past tense):

(108) A <COREF ID=60 REF=56 MIN="set">mesmerizing set</COREF>.

Some nominal constructions are intrinsically difficult for parsing and we cannot even expect a state-of-the-art parser to analyze them correctly:

(109) Anyone buying the basic package will receive <COREF ID=61 MIN= "stations">Cecchi Gori's three stations, <COREF ID=60 REF=61>Telemontecarlo, Telemontecarlo 2, and Videomusic</COREF>,</COREF> as well as BBC World, CNN International, Discovery, MTV Europe, Cartoon Network, and DMX music channel.

A parser would need a rather complex inference strategy to correctly analyze the noun phrases in (109), especially the noun phrase "Telemontecarlo, Telemontecarlo 2, and Videomusic". It should identify "Cecchi Gori's three stations" as a first part of an apposition, analyze its semantics, and then form the second part by conjoining exactly the *three* following names. We will see below (cf. Errors in Identifying Syntactic Indicators of Coreference) that such appositive-coordinate constructions are truly problematic for state-of-the-art parsers even if the markables themselves are extracted correctly.

Our NE-tagger (Curran and Clark, 2003b) relies on the MUC set of tags, distinguishing between PERSON, ORGANIZATION, LOCATION, MONEY, DATE, TIME, and PERCENT. It fails to recognize other types of entities, for example, PRODUCTs. This is partially compensated by the parser, as PRODUCT names are usually some NP-like constituents and therefore are included in the pool of markables (110A), unless they are modifiers (110B).

(110) A. "She loves <COREF ID=99>Chicken McNuggets</COREF>," Cook said...
B. Worried by information that China was seeking to buy <COREF ID=17>SS-18</COREF> strategic missile technology from Russia or Ukraine...

Our system has correctly analyzed "Chicken McNuggets" in (110A), but missed "SS-18" in (110B). More complex named entities, for example, TITLEs (of books, songs,...) are not covered by our preprocessing modules:

(111) Then a quick shift into "So Alive" – hey, this crowd was already pleased – and another cover, this of <COREF ID=34 MIN=""Rock On,"">David Essex's "Rock On,"</COREF> complete with a quip! "<COREF ID=33 REF=34 MIN="song">This song</COREF> will one day make David Essex's granddad rich," said one L&R.

If our NE-tagger had a better set of labels, we could have included "SS-18" and "Rock On" into our markables pool. It must however be noted that most state-of-the art NE-taggers support only very simple NE labelling schemes.

Incorrect markables extraction causes around 35% or our recall errors, making it the most common error type. We have to improve the reliability of our preprocessing modules in order to repair these errors: even with the best pos-

sible features and a perfect classification algorithm we cannot resolve anaphors not included in our pool of markables. State-of the-art parsing and NE-tagging modules (Charniak, 2000; Curran and Clark, 2003b) generally have a high performance level, but some specific constructions are still problematic. A possible remedy would be creating separate mini-parsers or taggers to cover difficult cases (for example, appositive-coordinate constructions).

Propagated Precision Errors. A markable and all its (manually annotated) antecedents may be correctly extracted by the preprocessing modules, but the system may still never see any positive testing instances and thus fail to resolve the anaphor:

(112) The company also said the Marine Corps has begun testing <COREF ID=19>two of its radars</COREF> as part of a short-range ballistic missile defense program. That testing could lead to an order for the <COREF ID=18 REF=19>radars</COREF> that could be worth between $60 million and $70 million.

Our preprocessing modules have suggested several candidate antecedents for "the radars": "The company", "the Marine Corps", "two", "its radars"... "order". The candidates have been submitted to the classifier one-by-one, starting from the closest markable ("order") and proceeding backwards. The classifier has correctly discarded most candidates, but then at some point has established a spurious link from "the radars" to "its radars". It has never seen any earlier markables, including the correct antecedent "two of its radars". In other words, a precision error has been propagated to decrease the system's recall.

This problem is crucial for anaphors that typically have their antecedents in the same or 1-2 preceding sentences – parts of appositive or copula constructions (113) and pronouns (114). If we incorrectly analyze the anaphor's context and extract erroneous values for syntactic or salience features, we are very likely to suggest an early spurious candidate:

(113) <COREF ID=66 MIN="Leonid Kuchma"><COREF ID=65 REF=66 MIN="president">The president of Ukraine</COREF>, Leonid Kuchma,</COREF> is the <COREF ID=68 REF=65 MIN="director"> former director of the factory in Ukraine that built the SS-18</COREF>.

(114) <COREF ID=14 REF=10>Silicon Graphics</COREF> no doubt hopes "FireWalker" will help jump start <COREF ID=16 REF=14>its</COREF> recent sluggish performance, but that's not guaranteed.

The parser has incorrectly attached "Leonid Kuchma" to "Ukraine" in (113), the classifier has linked these two markables (considering them parts

of an apposition) and has lost any chance to put "Leonid Kuchma" into the chain for "The president of Ukraine". The pronoun "its" in (114) has been resolved to the (incorrectly extracted) markable "jump" and the classifier has never seen its true antecedent, "Silicon Graphics".

Propagated precision errors account for around 7% of our recall errors. We see two possible remedies to the problem. One could improve the classifier, especially its precision level, to reduce the number of errors that could get propagated. Our examples show, however, that this is hardly a feasible solution: the erroneous links in (112) and (113) are very plausible and we cannot expect the improved classifier not suggest such links. One could alternatively refine the overall sampling strategy, forcing the classifier to check more markables even after some suitable antecedents had already been found[5].

Errors in Pronoun Resolution. Our system has failed to find a suitable antecedent for 91 pronoun. For 14 pronouns, the system has suggested a spurious antecedent too early and has never seen any positive testing instances (see Propagated Precision Errors above). Below we only discuss the remaining 77 cases, when the classifier has seen at least one positive testing instance, but has nevertheless failed to recognize it.

Table 8.6 shows the distribution of recall errors for different pronouns. The most problematic are first person plural (21 error), third person singular (28) and third person plural (20) pronouns. Our classifier mainly relies on salience features for pronoun resolution. The features already incorporate agreement constraints (number, person, and gender, see Section 6.6 for details). Our classifier often fails to suggest any suitable antecedent for first person plural pronouns. Too many candidates, on the contrary, are usually found suitable for third person pronouns and an incorrect one is finally picked.

First person pronouns are potentially difficult for any coreference resolution algorithm. They typically occur in embedded sub-discourses (see Section 6.3) and their resolution requires identifying the speaker of the embedded fragment and, for plural pronouns, reconstructing the represented group:

(115) Lancaster Rep. Robert Walker hopes to make space the private sector's newest frontier. "<COREF ID=14>Our</COREF> goal is to dominate both the commercial and government parts of space exploration," Walker said. The retiring Republican chairman of the House Committee on Science wants <COREF ID=13 REF=14>U.S.</COREF> businesses to compete in the commercial launch industry... (9 sentences)

[5]Similar approaches have been advocated in the literature. Ng and Cardie (2002c) have suggested to rate *all* the candidate antecedents according to the classifier's confidence value and then pick the best one. Harabagiu and Maiorano (1999) have investigated different processing orders (left-to-right vs. right-to-left for different sentences). In Section 6.8.1 we have evaluated window-based sampling for pronoun resolution.

> "<COREF ID=30 REF=13>We</COREF> need to make it easier for the private sector to compete in the space industry," Walker said.

This snippet illustrates the amount of reasoning and mere guessing we have to incorporate into our system to resolve first person pronouns. We first have to analyze the syntactic structure of the last sentence and compare it to the frame of the verb "say" (taken, for example, from the FrameNet data) to determine the speaker of the embedded fragment. It is not problematic for this particular example, but, nevertheless, requires additional external linguistic resources. We then need to link "Walker" to "Lancaster Rep. Robert Walker" to get at least some knowledge of this person. Finally, we have to reconstruct the group entity $we(Walker)$. This is the most difficult part, because $we(Walker)$ can occasionally refer to any group of people as long as it includes Robert Walker, ranging from his family to the whole mankind. The context suggests that "we" is probably somehow related to the politics, because the speaker was introduced as "Lancaster Representative". This consideration narrows down the set of candidates, but we still have no evidence to prefer, for example, the reading $we =$ "$U.S.''$ reading over another possibilities ($we =$ "$House\ Committee\ on\ Science''$,...).

Coreference resolution for first person pronouns, singular and plural, is a challenging and largely unexplored task for both shallow and deep approaches. We believe that it's an important open issues for future research. Our system has no knowledge to analyze such anaphors and therefore it has mostly suggested no antecedents. A few first persons pronouns have been resolved, usually incorrectly – the system overestimated the impact of matching features for pronominal anaphora (see Pronominal Anaphora in Section 8.2.2).

Third person pronouns are mainly resolved to very salient candidates. If the correct antecedent is not a salient entity, it is likely to be missed:

(116) But they agreed that the explanation for the <COREF ID=33 REF=29> History Channel</COREF>'s success begins with <COREF ID=34 REF=33>its</COREF> association with another channel owned by the same parent consortium.

The classifier has suggested a spurious antecedent, "the explanation" for the pronoun "its". The correct markable, "History Channel" has got a low salience value[6].

[6] "History Channel" is not the first NP in its sentence/paragraph, not a subject and not a CB. In Section 6.7 we have investigated a family of salience measures reflecting the discourse prominence of whole entities, as opposed to separate markables. "History Channel" is highly salient according to these measure (for example, it's re-mentioned many times in the document and can therefore be considered its topic). Our SVM classifier, however, has relied mainly on the markable-level salience features.

We can try and fix these errors by paying more attention to syntactic and semantic features and penalizing the classifier for making salience-based decisions for same-sentence pronominal anaphora. We can also rely on co-occurrence statistics, following the approach of Dagan and Itai (1990):

(117) And why not, since 75 percent of <COREF ID=28 REF=23>McDonald's </COREF> diners decide to eat at <COREF ID=29 REF=28>its</COREF> restaurants less than five minutes in advance?

The Google search engine provides 552.000 web pages mentioning "McDonald's restaurant" and no pages with "percent's restaurant". This clearly suggests a preference for "McDonald's" over "75 percent" (proposed by our system) as an antecedent for "its". The co-occurrence counts are, unfortunately, not always helpful:

(118) "The <COREF ID=75 REF=68 MIN="operator">cable operator</COREF> doesn't care how old <COREF ID=76 REF=43 MIN="subscriber"><COREF ID=77 REF=75>his</COREF> subscriber</COREF> is as long as <COREF ID=78 REF=76>he</COREF> pays <COREF ID=79 REF=78>his</COREF> monthly bill."

We have two morphologically plausible candidates for the pronoun "he" – "cable operator" and "his subscriber". The Google search engine provides very similar co-occurrence data for these candidates: 43.000 pages for "operator pays" (229.000.000 for "operator") vs. 38.400 for "subscriber pays" (262.000.000 for "subscriber"). These figures suggest, if any, a preference for the wrong candidate, "operator".

Intersentential anaphors are resolved more reliably. Our system picks the most salient candidate with the matching agreement values and this solution is usually accurate. The classifier still misses links with disagreeing markables:

(119) That means <COREF ID=46 MIN="mergers">cross-border mergers </COREF> are "quite possible," he said, "and in the end, I think <COREF ID=45 REF=46>it</COREF>'s mandatory."

Most state-of-the-art coreference resolution engines rely on strong agreement constraints and are therefore unable to link morphologically different markables. We think that a more detailed data-driven investigation of coreference links between morphologically different markables is required (see Errors in Nominal Anaphora Resolution below for a related discussion).

Finally, cataphoric pronouns may sometimes decrease the system's recall:

(120) LOVE & ROCKETS LAUNCH ANGST MINUS <COREF ID=26>

Pronouns (anaphor)	Errors		# pronouns in the data	
			anaphoric	total
1st sg	5	33.3%	15	15
1st pl	21	45.7%	46	49
2nd	0	0%	8	12
3rd sg: *(s)he,...*	6	12%	50	50
3rd sg: *it,...*	22	34.4%	64	88
3rd pl	20	34.5%	58	58
total	74	30.7%	241	272

Table 8.6: Recall errors for different types of pronominal anaphors on the testing data (20 MUC-7 "formal test" documents).

> MELODIES</COREF> ...
> What did they leave behind? Unfortunately, <COREF ID=25 REF=26> it</COREF>'s a <COREF ID=27 REF=25>biggie</COREF>: <COREF ID=28 REF=27 MIN="melodies">killer melodies</COREF>.

The (cataphoric) pronoun "It" is coreferent with "biggie". The latter participates in a coreference chain starting before the pronoun and therefore the annotators have to "resolve" the cataphoric "It" to some preceding markable[7]. Such cases cannot be covered by any right-to-left coreference resolution algorithm: there is no discourse clues in the document suggesting a relation between "MELODIES" and "It", so, the only way to establish this link is to resolve "It" to "biggie" (and "killer melodies") and then re-arrange the chain.

To summarize, around 16% of our recall errors are missed or incorrectly resolved pronouns. Our classifier relies mainly on salience features for pronominal anaphora resolution. This works well for intersentential links, but intrasentential coreference remains problematic. We should pay more attention to same-sentence pronominal anaphora resolution and increase the role of other linguistic factors. Pronouns are under-represented in our training data – only 418 markables (9%) in the 30 MUC-7 "dry-run" documents are pronouns (see Table 4.1). We hope to improve the system's performance by re-training the classifier on an external corpus annotated specifically for pronoun resolution (for example, Tetreault (2001)).

Errors in Name-Matching. Around 7% of our recall errors are matching mistakes: the classifier fails to link two variants of the same name. Missing links between two different proper names, referring to the same entity (for

[7]The MUC-7 guidelines explicitly state that the coreference relation to be marked is not directional, but in practice the links suggested by the annotators are always pointed from right to left.

example, metonymies) are discussed under Errors in Nominal Anaphora Resolution below.

Names of ORGANIZATION are the most difficult NE-anaphors for our system, contributing to 20 of 31 matching-related recall errors. Organizations are typically introduced by their official names and then further re-mentioned by simplified descriptions. The official variant may include the organization's name itself (for example, "Mild Seven Benetton Renault", "Blackwell", or "MTV"), followed by its domain ("F1", "Publishing"), its geographic descriptor ("International", "Europe"), and its legal form ("Corp.", "Ltd"). Simplified descriptions may include some parts of the full name ("Benetton", "Renault", and "Benetton Renault" are simplified versions of "Mild Seven Benetton Renault F1 Team"). Abbreviated versions of the official name (or its parts) may also serve as a simplified description.

In Chapter 3 we have introduced several simple features to deal with different variants of proper names. Especially important for names of ORGANIZATIONs are the `abbrev` and `first` matching strategies[8]. Although we have features to account for a number of NE simplification schemes, the classifier only learns very common matching patterns (for example, matching two markables with the same surface form or with the same head noun), missing more complex cases:

(121) CD RADIO STOCK RISES ON SPECULATION THAT <COREF ID=28>FCC</COREF> WILL GRANT LICENSE...
The Washington, D.C.-based company has waited for about six years to get a license from the <COREF ID=27 REF=2>Federal Communications Commission</COREF> to operate a system that would let customers tune into its radio stations anywhere in the country.

The NE-tagger has analyzed both "FCC" and "Federal Communication Commission" as names of ORGANIZATION and our `abbrev` features recognized an abbreviation here, but the classifier nevertheless failed to link the markables. Coreference links between a full and an abbreviated version of the same name are under-represented in our training corpus and therefore the learners cannot reliably extract them. We expect to get better results on abbreviations by adding more training material.

Some name-matching patterns are still not covered by our features:

(122) Softbank, which wholesales personal computer software and publishes personal computer-related materials, had purchased <COREF ID=73> Ziff-Davis Publishing Co.</COREF> last October for $2.1 billion.

[8] Our `first` matching functions compare the first words of two markables (with or without determiners, see Chapter 3 for details). The `abbrev` functions account for most common abbreviation patterns.

<COREF ID=72 REF=73>Ziff</COREF> is the <COREF ID=74 REF=72 MIN="publisher">U.S.'s largest publisher of computer magazines </COREF>.

The full description, "Ziff-Davis Publishing Co." contains, among other elements, two distinct names, "Ziff" and "Davis". The short description, however, contains only the first part, "Ziff". This is a common simplification pattern: for example, "Lockheed Martin Corporation" is often referred to as "Lockheed". Two parts of "Ziff-Davis", however, are conjoined with a hyphenation mark, and thus are not analyzed as single words and, consequently, can not be matched by our `first` functions.

Names of PERSON and LOCATION are matched more reliably: we have observed only 6 and 3 recall errors correspondingly. All the mis-matched PERSON names are of non-English origin. We have seen in Section 3.2 that such names can show very complex structure and therefore are extremely difficult to process:

(123) Ms Wu however acknowledged that China had suffered from a downturn of Taiwanese investment since cross-strait relations plummeted after <COREF ID=101 MIN="Lee Teng-hui">Taiwan President Lee Teng-hui</COREF>'s visit to the United States last June...
Both the People's Daily and Liberation Army Daily on Saturday carried a front-page editorial blaming the <COREF ID=100 REF=101 MIN="President">Taiwan President</COREF> for the current crisis. ¡p¿ The editorial claimed that Taiwan had plunged into chaos and the economy was on the brink of total collapse because of the separatist path trodden by <COREF ID=105 REF=100 MIN="Lee">President Lee</COREF>.

The only problematic LOCATION in the MUC-7 data is "United States of America", which has several variants, including abbreviated ("U.S.", "US", "USA", "U.S.A.") and simplified forms ("United States", "States")[9].

To summarize, our system has missed 31 NE-anaphor with at least one similar antecedent. We need a deeper NE analysis to resolve such links, distinguishing between different sub-parts of a proper name (for example GIVEN_NAME vs. FAMILY_NAME for PERSONs, or NAME, DOMAIN, etc for ORGANIZATIONs). Such functional sub-units of named entities are important for NE simplification patterns. State-of-the-art shallow methods for NE resolution, however, do not support these distinctions.

[9]Note that only "States" is a correct simplified form for "United States", whereas "United" is a different name, "United Airlines".

Errors in Identifying Syntactic Indicators of Coreference. We have seen in Section 4.7 that some syntactic constructions are very strong indicators for coreference, but they can often be confused with other syntactic structures and therefore require sophisticated extraction patterns, based on a parser's output. We have developed a set of heuristics for identifying appositions and copulas (see Section 4.7 for details). Incorrectly extracted or missed syntactic indicators for coreference account for 39 of our recall errors.

We identify candidates for appositions with a regular expression matcher, and then refine the candidate set, discarding, for example, addresses or coordinate constructions (see Section 4.7 for details). The regular expression is very simplistic and only matches comma-formed appositions ($[[.+]_{NP1}$, $[.+]_{NP2}$,?$]_{NP}$). Our system still misses appositions with other punctuation marks, for example, parentheses (124), and constructions with extra words ("or", "called", "known", etc) between two parts of an apposition (125):

(124) Softbank and News Corp. had already announced last Thursday a joint venture to purchase a <COREF ID=53 MIN="stake">20 percent stake in of Japan's TV Asahi broadcasting network</COREF> for <COREF ID=52 REF=53>41.75 billion yen</COREF> (<COREF ID=54 REF=52> $387 million</COREF>).

(125) "The downlink bands are not paired with any uplink bands," the company wrote. Indeed, for 1000 megahertz allocated for <COREF ID=93 MIN="downlinks">satellite downlinks</COREF>, or <COREF ID=92 REF=93 MIN="transmissions">transmissions from satellites to earth stations</COREF>, the agency had only set aside 500 megahertz for uplinks.

We have to incorporate additional regular expressions to identify such candidates. It requires more linguistics data annotated with appositive constructions or related information.

Some candidate appositions, for example, embedded in coordinate structures, are discarded to exclude syntactically similar constructions:

(126) Those materials, in turn, were encased in <COREF ID=68 MIN="Kevlar">Kevlar, a <COREF ID=67 REF=68 MIN="fiber">synthetic fiber </COREF>,</COREF> and Nomex to achieve a test strength of 400 pounds.

Such complex appositive-coordinate constructions are intrinsically problematic for parsing: a typical state-of-the-art parser has no knowledge that helps prefer the 2-entities interpretation ("[[Kevlar], [a synthetic fiber],] and [Nomex]") over the 3-entities interpretation ("[Kevlar], [a synthetic fiber], and [Nomex]").

The second part of an appositive construction may be a coordination:

(127) A. "It makes a biologist water at the mouth," said <COREF ID=24 MIN="C. Richard Tracy">Dr. C. Richard Tracy, <COREF ID=23 REF=24><COREF ID=25 REF=24 MIN="director">director of the Biological Resources Research Center at the University of Nevada at Reno</COREF> and <COREF ID=26 REF=24 MIN="member">a member of the Desert Tortoise Recovery Team, a group of researchers appointed by the U.S. Fish and Wildlife Service</COREF></COREF></COREF>.

The 23rd markable is a coordination and both its parts, "director" and "member", should be resolved to the 24th markable, "Dr. C. Richard Tracy", according to the manual annotation. Such links cannot be captured by our system. Moreover, they can hardy be captured by any automatic approach. On the one hand, we can assume that the pattern "$\texttt{NP}_1$`, `$\texttt{NP}_2$ `and` $\texttt{NP}_3$" is an indicator for a coreference chain $\{NP_1, NP_2, NP_3\}$ if the noun phrases are morphologically and semantically compatible. On the other hand, the pattern is sensible to parsing errors (and, as we have already seen, appositive-coordinate constructions are problematic for state-of-the-art parsers) and compatibility is difficult to determine (cf. Chapter 5). For example, the second appositive construction in (127), "the Desert Tortoise Recovery Team, a group of researchers appointed by the U.S. Fish and Wildlife Service" has been misanalyzed by the parser: "$[\text{the Desert Tortoise Recovery Team}]_1$, $[\text{a group of researchers appointed by the U.S. Fish}]_2$ and $[\text{Wildlife Service}]_3$". Our pattern would then assume coreference links between "the Desert Tortoise Recovery Team", "a group", and "Wildlife Service".

Examples (126) and (127) show that appositive-coordinate constructions are potentially difficult for any coreference resolution engine. First, parsers often analyze such sentences incorrectly. Second, it's difficult to partition extracted markables into coreference chains even for accurately parsed appositive-coordinate constructions. We have proposed special, more accurate features for appositions (see Section 4.7), discarding constructions containing a coordination or embedded in it. This helps to improve the system's precision at the expense of recall: currently our classifier treats appositive-coordinate constructions as mere coordinations and does not suggest any indicators for coreference in such cases.

Copula constructions are less problematic – our system has missed only 8 coreference links between a subject and its predicate nominal. In all these cases, the predicate is always a rather infrequent verb or verbal expression:

(128) Softbank president Masayoshi Son explained that the <COREF ID=22

REF=23>venture</COREF> will become synonymous with <COREF ID=24 REF=22 MIN="JSkyB">JSkyB, <COREF ID=25 REF=24 MIN="project">the 100-channel digital satellite project recently announced by News Corp.'s chairman and chief executive officer Rupert Murdoch </COREF></COREF>.

We have compiled a list of verbs and expressions that can potentially be used as predicates of a copula construction. Less common paraphrases, such as "become synonymous with" are not covered by our system.

We have only investigated appositive and copula constructions in our study. The MUC-7 data, however, suggest another coreference indicator based on syntactic and semantic properties of markables and their contexts. The annotators often mark as coreferent parts of constructions of the form ``$\mathtt{NP}_1$ PP $\mathtt{NP}_2$'', where one of the NPs denotes some quantity, price, etc:

(129) CD Radio stock rose 2 7/8 to 13 5/8 in trading of 400,300 shares, more than quadruple the <COREF ID=22 MIN="average">three-month daily average</COREF> of <COREF ID=21 REF=22 MIN="shares">88,700 shares</COREF>.

We do not propose any special rules or features for anaphoric links between parts of quantitative constructions: we believe that such structures can only be identified very unreliably, introducing too much noise. An accurate treatment of quantitative constructions requires much more data and we have only very few relevant examples in the MUC corpus.

To summarize, around 8% of our recall errors are caused by deficiencies in extracting and interpreting syntactic indicators for coreference. Around half of missed indicators are constructions not covered by our system: quantitatives, copulas with less common predicates, and appositions without commas or with extra words ("called", "or", etc). These errors can partially be repaired by refining extraction rules for our syntactic features. It must be noted, however, that the missing constructions are not very common and therefore it is unlikely that such refined extraction rules show a significant improvement on another testing corpus. The second half of errors are coreference links between parts of appositive-coordinate constructions. We have discarded such candidates to avoid precision losses, because our preprocessing modules cannot reliably analyze appositive-coordinate structures. The interaction of a coreference resolution engine with its preprocessing modules is an important issue discussed in Section 8.3 below.

Errors in Nominal Anaphora Resolution Below we focus on the major classes of errors in nominal anaphora resolution – missing links between non-pronominal anaphors and their MUC-7 antecedents. We assume that the

anaphor and at least one of its antecedents have been correctly extracted by our preprocessing modules, and that the markables are not variants of the same name and do not form an appositive or copula construction (we have already discussed problematic cases in markables' determination, name-matching and identifying syntactic indicators for coreference).

Nominal anaphora resolution is intrinsically difficult for an automated approach: different factors should be taken into account, including semantic compatibility of the markables' heads, their pre- and post-modifiers and the context. Our system can reliably resolve "easy" cases of coreference – links between NPs with the same head noun, missing only 4 such anaphors. Below we analyze more complex links. Our system has failed to establish 104 coreference links between markables with different head nouns: 63 links between two common noun phrases and 41 link involving at least one named entity.

The markables' heads may be morphological variants of the same lexeme. For example, singular noun phrases are often used to refer to group entities or classes of objects. Thus, "the desert tortoise" below refers not to a specific animal, but to the whole species:

(130) Hidden in <COREF ID=65 REF=64>their</COREF> burrows, <COREF ID=66 REF=65>desert tortoises</COREF> are perhaps as difficult to find as missiles stashed in underground silos...
"If you are going to manage the population and try to get the <COREF ID=67 REF=66>desert tortoise</COREF> to recover," said Steve Ahmann, the manager of natural and cultural resources for the Army at Fort Irwin, "it is really incumbent on you to know the population.

Our system has no information for detecting generic NPs, and therefore the classifier generally prohibits coreference links between markables having incompatible agreement features.

Some markables have very non-specific head nouns, that are semantically compatible with almost any other noun. In such cases, prenominal modifiers and embedded NPs often play a crucial role:

(131) "We believe that CNNsi will provide our subscribers an in-depth look at the <COREF ID=56 REF=9 MIN="stories">news stories that surround the competition and the athletes</COREF>," said Denny Wilkinson, Primestar's senior vice president of programming and marketing...
It will feature <COREF ID=76 REF=56 MIN="news">sports news </COREF> and talk 24 hours daily.

The markable "news stories" has a non-specific head "stories" and is potentially compatible with many entities in the domain of broadcasting, but its modifier, "news", could restrict the set of candidates and help link it to "sports

news". Our system (like other state-of-the-art coreference resolution engines) has virtually no features to compare the head of one markable to the modifiers of another one – their overlap is only reflected in the surface similarity values (various features encoding the minimum edit distance, see Chapter 3).

In our next example, the antecedent is a complex NP with the numeral "70" as its head word. We have to analyze two nested NPs to determine its proper meaning, "70 missiles", and establish a coreference link between "70" and "The missiles":

(132) With the Army contract, Lockheed Martin's Vought Systems unit, based in Dallas, begins production of <COREF ID=32 MIN="70">70 of the latest version of the newest version of the Army Tactical Missile System ground-attack missile</COREF>. The <COREF ID=31 REF=32> missiles</COREF> are supposed to be delivered by April 1998.

Our classifier has compared only the head nouns, missing the required link.

Coreference chains often include descriptions with semantically similar head nouns:

(133) As peaceful as that may seem, a report on the <COREF ID=7 REF=3> satellites</COREF>' findings, completed in March, was designated as secret because the information could reveal too much about the abilities of <COREF ID=10 REF=7 STATUS=OPT>U.S. reconnaissance technology</COREF>...
Rather, the study's importance lay in the use of <COREF ID=17 REF=10 MIN="tools">advanced intelligence-gathering tools</COREF> to examine the environment, an application that scientists say has enormous potential benefits for future research. <COREF ID=21 REF=17 MIN= "instruments">Remote-sensing instruments</COREF> could save time and money in various projects, producing data that would otherwise be hard to gather.

We have seen in Chapter 5 that determining semantic compatibility of two nouns is an extremely difficult task. Figure 8.2 shows the WordNet subtree for the first senses of the head nouns in (133), "satellites", "tools", "instruments", and "technology". The noun *technology*, according to the WordNet tree, is very different from the other three candidates, belonging even to another top node – *act* vs. *entity*. The nouns "satellite", "tool", and "instrument" have a high WordNet similarity rate (see Section 5.3 for an overview of similarity metrics used in our study). This however does not necessary mean that they are compatible: for example, another descendants of *instrumentality*, such as *dispenser* (Figure 8.2, dotted line), are also very similar to *satellite* and *tool*, but only compatible with the latter one. We have also investigated another way of as-

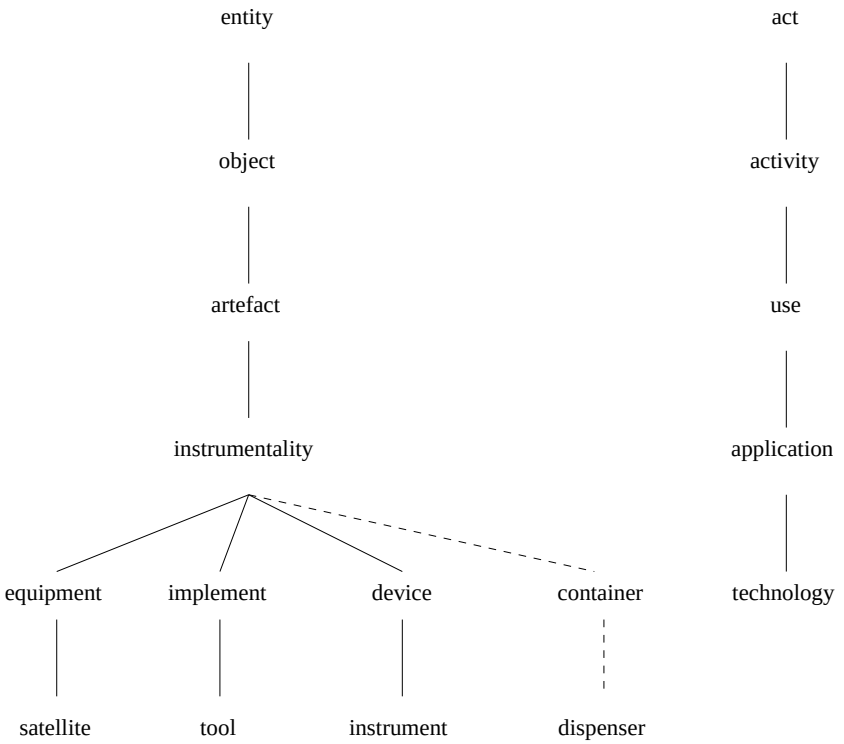


Figure 8.2: Fragment of the WordNet hierarchy for Example (133).

sessing semantic compatibility, encoding paths connecting two WordNet nodes (for example *satellite*→*equipment*→*instrumentality*←*device*←*instrument*, see Section 5.4 for details). This method either cannot help us distinguish pairs of compatible nouns (*satellite*, *tool*) from pairs of incompatible nouns (*satellite*, *dispenser*).

Coreferring markables with different head nouns may also disagree in number. Thus, a singular NP "spy craft" below refers to an entity mentioned earlier as "spy satellites" and "they" (note, however, that the MUC-7 analysis of this snippet is questionable – the first two entities are probably not generic but rather refer to some specific satellites):

(134) In the Northwest, <COREF ID=41 REF=7 MIN="satellites">spy satellites</COREF> have gauged the temperature of salmon streams. In Alaska and Florida, <COREF ID=42 REF=41>they</COREF> have helped map wetlands...
<COREF ID=49 REF=42 MIN="craft">Spy craft</COREF> can examine Earth in great detail using telescopic cameras and dozens of electromagnetic wavelengths.

Most coreference resolution systems filter out candidates with mismatching number or gender values. Cristea et al. (2002) have suggested to explore WordNet glosses to identify group entities, such as "the patrol". This would help to resolve *some* anaphors with morphologically incompatible antecedents. Many other links, however, would remain problematic, including local topics (consider a singular pronominal anaphor "it" with a plural antecedent in Example (117)) or generic descriptions ("the desert tortoise" in (130)). We believe that an extensive corpus-based analysis is needed to investigate the types of links violating agreement constraint and to develop a full-scale algorithm for their resolution.

We rely on external resources (WordNet) for comparing different common names. Computing semantic compatibility from the WordNet data is a very non-trivial problem that cannot be resolved efficiently at the moment. Nevertheless, we still can hope to improve our techniques and automatically extract more reliable information from external sources. This would help us to resolve common noun phrases, but named entities would still remain problematic. Our system has missed 41 coreference link with at least one proper name. Example (135) illustrates the most common problem with NE-antecedents:

(135) <COREF ID=27 REF=20>Aerospatiale</COREF> has already taken some steps in that direction. Several years ago <COREF ID=28 REF=27> it</COREF> merged <COREF ID=29 REF=28>its</COREF> helicopter operations with those of Daimler Benz, forming Eurocopter, and the <COREF ID=32 REF=29 MIN="company>French company</COREF> plans to merge both <COREF ID=33 REF=32>its</COREF> satellite and missiles operations with those of Daimler Benz by September.

This snippet contains 3 plausible candidates for "the French company": "Aerospatiale", "Daimler Benz", and "Eurocopter" (note that we already need some knowledge to exclude such markables, as, for example, "that direction," from the candidate pool). The "Daimler Benz" solution could be ruled out after a deep analysis of the sentence: the reading "Daimler Benz plans to merge both its satellite and missiles operations with those of Daimler Benz by September" is not plausible. Such analysis, however, relies on a lot of knowledge and can hardly be achieved using state-of-the-art NLP resources. And even such advanced technologies could not help us exclude "Eurocopter" from the candidate set.

An alternative approach would require a knowledge base mapping each company to its primary location. We could consult the database to find out that "Aerospatiale" is indeed a French company and "Daimler Benz" is not. "Eurocopter", however, would probably remain problematic – no conventional size database could possibly contain names of numerous spin-offs and young

companies.

Finally, another potential analysis for this particular problem relies on an external "origin-guessing" module – given a proper name, we can try and guess its (linguistic) origin and thus determine the companies location ("Aerospatiale" is French, "Daimler Benz" – German, and "Eurocopter", again, problematic). Although state-of-the-art algorithms can reliably identify the language of even a small document, stand-alone names remain problematic (see, for example, (Llytjós, 2002)). The underlying assumption "location = linguistic origin" is also very questionable.

Summarizing, we have outlined three different processing strategies for coreference links involving the named entities in (135): deep sentence analysis, knowledge-based and statistic surface-based processing. All these solutions, however, rely on a lot of reasoning and are very specific: we have to analyze the entities' location to establish the correct links in (135), but we may need very different kinds of knowledge for another examples. This makes us believe that coreference resolution for NEs, apart from name-matching, can hardly be achieved with shallow methods.

Some named entities, especially DATEs, need a corpus-specific information for their resolution:

(136) <DATE> <COREF ID=7>06-25</COREF> </DATE> ...
<TEXT>
... That meeting eventually led to <COREF ID=75 REF=6>today</COREF>'s announcement, Son said, as the two companies realized their common interest in digital satellite broadcasting in Japan...
</TEXT>

There is no information in the text body that would help us determine the day of the document's creation. This knowledge, however, can be extracted from the DATE field. We should analyze the document's structure (see Section 6.2) to link all the subsequent mentions of the same date (e.g. "June 25", "today"). Such anaphors are "situationally evoked" (Prince, 1981) – when the reader sees "today" in a newspaper, she can easily interpret it relying on the situational context. The MUC-7 guidelines restrict the scope of coreference relations to "IDENT" – in Prince's (1981) terms, links between "textually evoked" anaphors and their antecedents. The guidelines explain that IDENT is the only relation that can be annotated reliably at the moment[10]. Such anaphors as "today"/"June 25" do not participate in the IDENT relation in the proper sense: they are rather situationally evoked and refer to the same

[10]This view is partially supported by Poesio and Vieira (1998): their annotators could only achieve a moderate agreement level on bridging anaphora.

date and, therefore, form a coreference chain by transitivity.

Coreference links between a common name (antecedent) and a proper name (anaphor) are often even more difficult. We have to rely on data-specific heuristics (in cases of contextually evoked descriptions similar to (136)) or very sophisticated inference to resolve such anaphors:

(137) WASHINGTON, Feb. 26 – Lancaster Rep. Robert Walker hopes to make space the private sector's newest frontier.
"<COREF ID=14>Our</COREF> goal is to dominate both the commercial and government parts of space exploration," Walker said.
The retiring Republican chairman of the House Committee on Science wants <COREF ID=13 REF=14>U.S.</COREF> businesses to compete in the commercial launch industry.

The resolution strategy for "U.S" in (137) involves reconstructing the entity for the pronoun "Our" (note that "Our" is a discourse new description here, so we have to truly reconstruct its referent from scratch, not relying on any already established coreference links) and linking the two markables. The first part is problematic (see Errors in Pronoun Resolution). Common noun antecedents ("the country", "the nation") would be easier, but even they require substantial processing. Finally, discourse entities are seldom introduced by common names or pronouns and then further re-mentioned as proper names. This consideration makes "U.S." an unlikely anaphor.

Coreference links between two different proper names are less common. Our system has missed 6 NE-NE links:

(138) But the officials said they had press reports and intelligence information that <COREF ID=33 REF=20>China</COREF> was shopping around for the massive, multiple-warhead SS-18s, with <COREF ID=36 REF=33>Beijing</COREF> seeking to purchase the missiles and components from cash-starved Russian and Ukrainian companies purportedly for <COREF ID=40 REF=36>its</COREF> civilian space-launching program.

Two distinct proper names, "China" and "Beijing", are used here metonymically to refer to the same object, "Chinese administration." We refer the reader to Markert and Nissim (2003) for computational account of metonymies.

Altogether, around 20% of the recall errors made by our system are complex cases of nominal anaphora. Their resolution requires some measure of semantic compatibility but, as we have seen in Chapter 5, even assessing compatibility for head nouns is a truly challenging problem. This makes nominal anaphora resolution an open issue for future research.

To summarize, our SVM^{light}classifier has the recall level of 63.9%, miss-

	Errors	%
MUC-7 inconsistencies	30	7.4%
Misleading markables	76	18.6%
preamble	24	5.9%
text body	52	12.7%
Pronoun resolution	78	19.1%
NE-matching	20	4.9%
Syntactic constructions	22	5.4%
apposition	12	2.9%
copula	10	2.5%
NP-anaphora	182	44.6%
multi-word expressions	3	0.7%
homonymy	4	1.0%
new modifier (anaphor)	15	3.7%
incompatible modifiers	30	7.4%
compatible modifiers	58	14.2%
no modifiers	62	15.2%
total	408	100%

Table 8.7: Performance of the SVM^{light}classifier: Precision errors on the testing data (20 MUC-7 "formal test" documents).

ing 469 anaphors. The most problematic are markables determination (166 errors) and nominal anaphors (108). We will discuss possible directions for improvement in Section 8.3 below.

8.2.2 Precision Errors

Precision errors occur when a system suggests coreference links not supported by the manually annotated corpus data: either at least one of the markables is incorrect, or they belong to different chains. Our SVM^{light}classifier has made 408 precision errors (Table 8.7).

In all the examples in this section, we use square brackets with lower indices to show (mostly erroneous) coreference links proposed by our system. We only show the anaphor under discussion and its chain.

MUC-7 Inconsistencies. Some links suggested by our system are missing in the MUC-7 corpus for no clear reason, which inevitably happens with a human-annotated resource. For example, "It" in (139) below is not marked in the MUC data, although it unambiguously refers to "Primestar":

(139) [Primestar]$_1$, [a cable industry consortium]$_{2,ante=1}$, offers some 95 channels. [It]$_{3,ante=2}$ has 1.3 million subscribers, who receive programming through a dish the size of a pizza pan.

Some anaphors are annotated inconsistently. Thus, the instances of "1998" have been manually annotated as coreferent in (140A), but not coreferent in (140B):

(140) A. A pure net profit, he said, may be out of reach until [1998]$_1$ now, though...
"With the U.S. dollar at five francs, we have to be profitable, at least in [1998]$_{2,ante=1}$."
B. But with no customers expected until [1998]$_1$, the need for nearly $2 billion in investment and numerous competitors lurking in the shadows, Globalstar's prospects would not appear to be valuable to the average Lockheed shareholder...
If Globalstar begins its service on schedule in [1998]$_{2,ante=1}$, he predicted that the company would have 3 million customers by 2,002, bringing in $2.7 billion in annual revenue.

According to the MUC-7 annotation guidelines, atomic date expressions (i.e. both mentions of "1998" in (140), but not "1998" in "Sept. 07, 1998") should be always marked. Our system consistently classifies two mentions of the same year (provided they are markables, that is, not parts of embedding date expressions) as coreferent, leading to a precision loss in cases similar to (140B).

Coordinate constructions are also sometimes marked inconsistently:

(141) Any such sale would violate treaty obligations on non-proliferation, including the first Strategic Arms Reduction Treaty, Start I, that [both Moscow and Kiev]$_1$ have ratified...
He told The Washington Times that the protests to China were "more general," but that the messages to [Moscow and Kiev]$_{2,ante=1}$ were "very specific on action they might take," presumably to control the companies or officials who may have been negotiating with Beijing.

The annotators have suggested two markables ("Moscow" and "Kiev") for the first coordination, but three markables ("Moscow", Kiev", "Moscow and Kiev") for the second coordination. The latter markable, "Moscow and Kiev" is the antecedent for the pronoun "they"[11]. Our system has identified as

[11]Note that these sentences are taken from the same document. If we assume that "Moscow and Kiev" is a discourse entity (re-mentioned as "they"), we should also annotate the first coordination, "both Moscow and Kiev."

markables and linked into a chain both coordinations.

The bracketing suggested by the annotators sometimes deviates from the guidelines:

(142) What really made the evening, though, was the opening set by the Oregon-based quartet [the Dandy Warhols]$_1$. The name might be silly but the band isn't...
Guitarist Courtney Taylor and [synthist Zia McCabe]$_2$ meshed beautifully – complex harmonics shimmering off simple chords – the highlight being "It's A Fast-Driving Rave-Up With [the Dandy Warhols]$_{3,ante=1}$' Sixteen Minutes." No kidding, that's the song title: The 16 minutes is a play on Andy Warhol's "15 minutes of fame" cliche, and the idea (as [McCabe]$_{4,ante=2}$ explained post-set) was "to have more words in the title than in the song."

The annotators have suggested a minimalistic bracketing "Zia McCabe" for the second markable (see also Example (97) above). Our system has proposed a correct chain {"synthist Zia McCabe", "McCabe"}, but the scoring program could not align it with the manually annotated data. The third markable, "the Dandy Warhols" has been manually annotated as "Warhols" and cannot be aligned either. This constituent, however, should not be marked at all, as it is a part of a larger name, "It's A Fast-Driving Rave-Up With the Dandy Warhols' Sixteen Minutes."

We have already mentioned (see "MUC-7 Inconsistencies" in Section 8.2.1) that the scoring program fails to analyze a markable, if it is embedded in a larger one with the same or intersecting "MIN" argument:

(143) BC-[SOFTBANK]$_1$-NEWSCORP-ALLIANCE-BLOOM
[SOFTBANK]$_{2,ante=1}$, NEWS CORP. FOCUS ON SATELLITE BROADCASTING (UPDATE2)

Our system has proposed a correct link from "SOFTBANK$_2$" to "SOFTBANK$_1$". The scorer, however, has aligned our second markable with the embedding NP, "SOFTBANK, NEWS CORP.", considering the suggested link erroneous and the second mention of "SOFTBANK" unresolved, which resulted in two errors.

To summarize, inconsistencies in the manually annotated testing material account for around 7% of our precision errors and 4% of the recall errors. This shows that the MUC-7 corpus is a relatively noisy dataset. The inconsistencies decrease the performance figures for the testing data, but also deteriorate the quality of the training material, making it difficult for machine learners to obtain a high-quality classifier.

Errors in Markables' Determination. Our system has suggested 76 markables not corresponding to any linguistically plausible constituent. They reflect deficiencies of our preprocessing modules. The system has also suggested several linguistically well-formed descriptions, not shown in the MUC-7 data – according to the guidelines, an NP should not be annotated if it does not participate in any coreference relation. We only discuss the former type of errors in this section – as we have seen in Chapter 7, determining, if an NP belongs to some coreference chain (and, thus, is a MUC-markable) is a challenging task of its own.

Auxiliary parts of a document (SLUG, DATE, NWORDS, and PREAMBLE) are problematic for our system:

(144) <SLUG fv=taf z> BC-[LORAL]$_1$-[SPACE]$_2$-470&A

"LORAL" and "SPACE", suggested as two separate markables, are in fact parts of a single description, "LORAL-SPACE" (see Example (102) above for the MUC-7 annotation of the same snippet). More accurate segmentation of auxiliary New York Times information is out of the thesis' scope.

Some markables suggested by our system are parts of longer proper names (145-147) or other NE-like fixed expressions (148):

(145) The plans are significant, said Scott Blake Harris, former [FCC] international bureau chief, as "yet another indication of the health and strength of the U.S. satellite industry."

(146) c.1996 South [China] Morning Post

(147) They shunned a set list, chucked most of their pop songs (save "[Nothing] to Do") to the side and launched into some monstrous ebb-and-flow, synth/guitar jams worthy of early Pink Floyd, Hawkwind or even Roxy Music.

(148) These talks must be conducted under the principle of "one [China]" and conducted in "proper capacity, appropriate time and conditions", she said.

Our NE-tagger has missed the "FCC international bureau" name, suggesting that "FCC" alone is a named entity (145). In (146), the embedding name, "South China Morning Post", belongs to a non-supported NE-type[12] and thus can not be analyzed correctly. Examples (147) and (148) are even trickier:

[12]The NE-tagger used in our experiments (Curran and Clark, 2003b) relies on the MUC classification of NEs, distinguishing between PERSON, ORGANIZATION, LOCATION, DATE, TIME, PERCENT, and MONEY.

although "Nothing to Do" and "one China" are, in some sense, named entities as well, such names can hardly be covered by any possible NE-classification scheme.

Parsing errors may also lead to spurious candidates:

(149) The board found that the tether met [all the mission]'s specifications for strength, temperature and electrical performance but noted that there was no requirement for the tether to be manufactured in a contaminant-free environment.

The parser has suggested an incorrect bracketing ("[[all the mission]'s specifications]" instead of "[all [the mission]'s specifications]"), leading to a spurious markable, "all the mission" and decreasing the system's precision.

To summarize, around 35% of our recall errors and 20% of our precision errors are due to missing/spurious markables. We have followed state-of-the-art systems in paying more attention to the links and simply collecting the markables by merging the output of general-purpose external modules. Our error analysis suggests, however, that we should elaborate our techniques for extracting markables. We have proposed a step in this direction in Chapter 7: we use machine learning to automatically discard unlikely anaphors and antecedents, including parsing errors, from the pool of candidates. We believe that we can significantly increase the system's performance level by improving its interaction with the preprocessing modules and therefore obtaining better markables.

Pronominal Anaphora. Our system has suggested spurious antecedents for 78 pronominal anaphors. Table 8.8 shows the distribution of errors for different types of pronouns. The most problematic are 1st person plural (9 errors), third person singular (48) and third person plural (17) pronouns. We have identified three major types of errors: wrong parsing/tagging, over-estimating the impact of matching features, and incorrect preference for salient same-sentence candidates[13].

Deficiencies of our preprocessing modules may lead to incorrect agreement values for some markables:

(150) [Two key vice presidents]$_1$, [Wei Yen]$_{2,ante=1}$ and Eric Carlson, are leaving to start [their]$_{3,ante=2}$ own Silicon Valley companies, sources said.

[13]We do not identify discourse new (pleonastic, cataphoric, etc) pronouns as a separate error class. Our system is not tuned to resolve *every* pronoun it encounters, so, if it suggests some (spurious) antecedent for a discourse new pronoun, we still can and have to investigate the error source. Our classifier has correctly left unresolved 10 (of 31) discourse new pronouns.

Pronouns (anaphor)	anaphoric			discourse new		
	Errors		# pronouns	Errors		# pronouns
1st sg	0		15	-		0
1st pl	8	17.4%	46	1	33.3%	3
2nd	1	12.5%	8	3	75%	4
3rd sg	31	27.2%	114	17	70.8%	24
3rd pl	17	29.3%	58	-		0
total	57	23.7%	241	21	67.7%	31

Table 8.8: Precision errors for different types of pronominal anaphors on the testing data (20 MUC-7 "formal test" documents).

The parser has analyzed "Wei Yen" as a plural NP (probably considering "Yen" to be the Japanese currency). This has made "Wei Yen" a plausible candidate for the plural pronoun "their".

Our system considers coreferent any two markables with the same surface form. This is generally a reliable resolution strategy (see Table 8.1 for the performance figures of the `same_surface` baseline). Pronouns, however, should be matched much more cautiously: the same pronoun may refer to distinct entities in different parts of a document. Over-estimating the importance of matching features therefore inevitably leads to spurious links: our classifier first fails to suggest a suitable antecedent for a pronoun (either because it's not anaphoric, or because the system misses the true candidate), and then it finds another pronoun with exactly the same surface form and links the two.

Inappropriate matching is the main source of errors for first person pronouns:

(151) They went on hiatus following an improbable $[\text{US}]_1$ hit, "So Alive," and were playing the summer shed circuit...
"This song will one day make David Essex's granddad rich," said one L&R.
"And $[\text{us}]_{2,ante=1}$ as well," said another.

The classifier has failed to resolve "us" to "L&R" and matched it to "US".

We could try and repair this errors by increasing the system's recall. We have, however, seen in Section 8.2.1 that first person pronouns are truly challenging for any coreference resolution algorithm and we can hardly expect significant recall improvements here. Even if we could fix *all* the recall errors made by our system, we would still obtain spurious links for discourse new pronouns:

(152) Silicon Graphics no doubt hopes "FireWalker" will help $[\text{jump}]_1$ start

> [its]$_{2,ante=1}$ recent sluggish performance, but that's not guaranteed...
> In [its]$_{3,ante=2}$ first week, the video game "Mortal Combat II" grossed $50 million – more than the movie "Forrest Gump" or "Lion King."

The third markable is a cataphoric pronoun and it should ideally be left unresolved (recall that we only attempt to link anaphors to preceding markables, so we have no possibility to resolve "its" to "Mortal Combat II"). Our classifier, however, has suggested a matching pronoun, creating an erroneous chain.

We also cannot solve the problem by extending our training corpus. The learner over-estimates the impact of matching features because it sees only extremely few examples of same-surface non-coreferring pronouns – they are mostly discarded by our sample selection algorithm (cf. Section 2.3). Training instances similar to {"US", "us"} in (151) occur very seldom: for example, if the snippet (151) was a part of a training document, it would produce just one training instance, {"L&R", "us"}, for the pronoun "us". Discourse new pronouns are never selected as anaphors to make training instances. We see two possible remedies to the problem: splitting matching features for pronominal vs. non-pronominal anaphors (see (Ng and Cardie, 2002c) for a similar solution) or changing the sampling strategy.

The last group of errors are pronouns, incorrectly resolved to a salient same-sentence antecedent. Our classifier generally suggests very salient candidates (see Sections 6.6 and 6.7 for a description of implemented salience criteria). This works well for intersentential anaphora; but the performance figures for same-sentence coreference are only moderate – in fact, 29 of the 78 precision errors are spurious intrasentential links.

Salience-based approaches to pronoun resolution typically account only for intersentential coreference. Tetreault (2001) has shown that most centering algorithms cannot outperform the naive syntactic approach of Hobbs (1978), if they are not extended to cover same-sentence anaphors. Our classifier has committed a similar mistake: it has not learned any specific (syntactic) patterns for intrasentential pronominal anaphora and has instead relied on salience features:

(153) Still, Son emphasized that [the venture's relationship]$_1$ with TV Asahi will not restrict [it]$_{2,ante=1}$ from dealing with other content providers.

The classifier has picked a salient subject NP, "the venture's relationship" as an antecedent, although our syntactic features (especially commands, see Section 4.6) could have discouraged the system from suggesting such links.

Salience parameters of entities are related to discourse structure and text coherence. They determine pronominalization strategies across utterances and are much less relevant for same-sentence coreference. This makes intra- and intersentential coreference very different tasks requiring two separate resolution

strategies and, possibly, two separate feature sets.

To summarize, inaccurate pronoun resolution accounts for around 20% of our precision errors. The most common error subtypes are misleading syntactic analysis, inappropriate matching and over-estimating the impact of salience features. The latter two groups are very pronoun-specific and can potentially be eliminated by re-training the classifier on a dataset annotated for pronominal anaphors. Our present classifier, with its uniform processing strategy for all types of markables, cannot prevent "pronominal matching" and distinguish between intra- and intersentential anaphora.

Errors in Name-matching. Newswire documents often describe distinct entities with similar names – relatives (PERSON) or spin-off companies (ORGANIZATION). The snippet below mentions "Loral", "Loral Space and Communications Corp.", "Loral Space", and "Space Systems Loral":

(154) News of Monday's deal, in which Lockheed will buy most of $[\text{Loral}]_1$'s military businesses and invest $344 million in [Loral Space and Communications Corp.]$_2$, [a new company]$_{3,ante=2}$ whose principal holding will be $[\text{Loral}]_{4,ante=1}$'s interest in , sent Globalstar's own shares soaring $6.375, to $40.50 in Nasdaq trading...
In addition, Schwartz said $[\text{Loral Space}]_5$ would use its holdings in $[\text{Space Systems Loral}]_{6,ante=4}$, [a private maker]$_{7,ante=6}$ of satellites, to expand into the direct broadcast satellite business.

It is difficult even for a human reader to correctly cluster these names: "Loral", "Loral Space and Communications Corp.", and "Space Systems Loral" are different companies, whereas "Loral Space" is another name for "Loral Space and Communications Corp".

Two similar names may refer to the same entity ("William Gates" and "Bill Gates"), distinct related entities ("Bill Clinton", "Hillary Clinton"), or unrelated entities ("Republic of China", "China, Mexico"). A name may also be misspelled (see Table 3.1 for a variety of queries for (presumably) "Qaddafi" and "Britney Spears" submitted to the Google search engine) or created explicitly to resemble a popular name ("Adadis" and "Adidas"). Distinguishing between similar names is a well-known problem in data mining, especially in database management (Borgman and Siegfried, 1992). We refer the reader to Section 3.1 for an overview of related studies.

Another problem arises with names of ORGANIZATIONs. Many companies are named after people who still play an important role in their administration:

(155) TELEPIU, $[\text{CECCHI GORI}]_1$ UNVEIL NINE-CHANNEL DIGITAL TV PACKAGE...

> Telepiu SpA and [Cecchi Gori Group]$_{2,ante=1}$ unveiled a nine-channel package of digital pay-television programming for Italy and said they will sign up four more channels in the next few weeks. ... [Vittorio Cecchi Gori]$_{3,ante=1}$ is [a Rome-based film producer]$_{4,ante=3}$ who is turning his Telemontecarlo and Videomusic television stations into a national network.

The first markable, "CECCHI GORI" is ambiguous – it can either refer to a company ("Cecchi Gori Group", the 3rd markable) or to a person ("Vittorio Cecchi Gori", the 2nd markable). Our system has independently resolved the second and the third markable to the first one, merging two coreference chains. Such errors inevitably occur if we process anaphors independently and could only be avoided by switching to a more global chain-level model of coreference. It must be noted, however, that this particular example contains a truly ambiguous description ("CECCHI GORI") and we cannot possibly decide which of the proposed two links should be eliminated. This is a problem of the "coreference as equivalence" approach advocated by the MUC-7 guidelines.

To summarize, deficiencies in our name-matching techniques contribute to 7% of recall and 5% of precision errors, making it the smallest error group. Even a naive matching baseline (`same_surface_normalized`) achieves a performance level of 53.7% (cf Table 8.1). Our more sophisticated matching features help us resolve more anaphors. The remaining mistakes are intrinsically difficult name-matching problems, requiring deeper NE analysis – identifying name structure, distinguishing, for example, between GIVEN_NAME and FAMILY_NAME for PERSONs. This task lies out of the scope of this thesis.

Syntactic indicators for coreference. Our classifier relies on two syntactic indicators for coreference – apposition and copula. Both constructions are extracted fully automatically. Parsing and NE-tagging errors or deficiencies of our extraction rules may result in incorrect values for our syntactic features, decreasing the system's precision level:

(156) [Son]$_1$ accompanied at the press conference by News Corp.'s Japan representative, [John McBride]$_{2,ante=1}$, on the eve of Softbank's annual shareholders meeting, said the terms of the joint venture, including capital and personnel issues, will be clarified in meetings with Murdoch next month.

(157) Since their evidence resulted in the government recovering money, the False Claims Act law says [Aldred and Goodearl]$_1$ are due [part]$_{2,ante=1}$ of the fine.

The parser has considered two NPs, "Son accompanied at the press conference by News Corp.'s Japan representative" and "John McBride", to be parts of an apposition. This parse has been reflected in our feature vectors and the classifier has finally put "Son" and "John McBride" into the same chain (159). Our system has incorrectly interpreted the parse tree for (157), considering "part" to be a predicate nominal and therefore resolving it to "Aldred and Goodearl".

We have discussed (see Section 4.7) the most common syntactic constructions having the same surface structure as appositions and copulas and adjusted our feature set to get more precise syntactic indicators for coreference, introducing more sophisticated features. We exclude, for example, appositions containing an embedded coordination and vice versa. Very complex cases of appositive-coordinate constructions are still sometimes analyzed incorrectly:

(158) Anyone buying the basic package will receive Cecchi Gori's three stations, Telemontecarlo, Telemontecarlo 2, and Videomusic, as well as [BBC World]$_1$, CNN International, Discovery, MTV Europe, [Cartoon Network]$_{2,ante=1}$, and DMX music channel.

The parse tree for (158) contains multiple embedded noun phrases, some of them bracketed incorrectly. Our system has failed to extract values for the relevant syntactic features from this complex tree.

Some syntactic structures, similar to appositions or copulas, are still not excluded. For example, topic constructions are erroneously analyzed as copula:

(159) [It]$_1$ is [the latter capability]$_{2,ante=1}$ that has military minds excited.

The (non-referential) pronoun "It" is used to promote "the latter capability" to the topic position and should not participate in any coreference chains.

To summarize, incorrect syntactic indicators for coreference account for around 8% of our recall and 5% of our precision errors. We have to improve the underlying parsing algorithm and our feature extraction rules for appositions and copulas to increase the performance level.

Errors in Nominal Anaphora Resolution. Around half of the precision errors made by our system are incorrectly resolved full noun phrases. Our classifier mainly relies on the `same_head` features for nominal anaphora resolution. Although it is generally a reliable strategy (cf. Table 8.1, the `same_head_normalized` baseline), the system has suggested 182 spurious links.

It is generally assumed in the literature (Poesio and Vieira 1998, among others) that one should pay closer attention to (pre-) modifiers to determine whether two same-head NPs are coreferent: for example, "the state-owned French companies" and "U.S. companies" below can hardly refer to the same

object, because "French" and "U.S." are incompatible:

(160) While [the state-owned French companies']$_1$ rivals across the Atlantic have been "extremely impressive and fast" about coming together in mergers, [European companies]$_{2,ante=1}$, hobbled by political squabbling and red tape, have lagged behind, Gallois said...
The competition is even tougher for Aerospatiale in that the U.S. dollar has weakened 10 percent against the French franc last year, giving [U.S. companies]$_{3,ante=2}$ what Gallois called a "superficial" advantage.

The bottleneck of this approach lies in the lack of required knowledge bases: we can compile small lists of mutually incompatible properties, but a large-scale general-purpose resource can hardly be produced manually in any reasonable time. The same properties can be expressed differently, not involving pre- or post-nominal modification. For example, the NP "Some companies" in "Some companies have moved their production to foreign companies (taking US jobs with them)" clearly refers to "US companies" and therefore can hardly be coreferent with "French companies". We have seen in Chapter 5 that it is extremely difficult to assess the semantic compatibility of two head nouns. It is obviously even more difficult to determine the compatibility of two "properties" that can be expressed in very (syntactically) different ways. We have performed a detailed analysis of the errors made by our system to see if (prenominal) modifiers could help avoid spurious links and, if not, what other clues should be used instead.

Two markables may have homonymous head nouns. We have found four cases of a pure homonymy and three cases of multi-word expressions:

(161) Turner Broadcasting System Inc., for [its part]$_1$, agreed in July to distribute Cable News Network and three other cable channels to Latin American subscribers together with a group called Galaxy Latin America, composed of GM's DirecTV, Venezuela's Cisneros Group of Cos., Brazil's Televisao Abril, and Mexico's MVS Multivision...
Those functions are likely to be slowly shifted to another slice of spectrum, while the airwaves they've historically used are turned over, in [part]$_{2,ante=1}$, to satellite services such as the ones planned by GE and GM.

We can potentially fix these errors by incorporating external modules for extracting collocations and word sense disambiguation. Such links, however, are rather uncommon, accounting only for around 1.5% of all our recall errors.

The remaining 175 spurious links are pairs of noun phrases with exactly the same head noun. We have to pay attention to other words from the markables themselves, their contexts, and their chains to rule out such links.

Some modifiers, for example, numeric expressions or possessives, are clearly incompatible:

(162) [Softbank's stock]$_1$ fell 300 yen to 19,500 yen. [News Corp.'s stock]$_{2,ante=1}$ rose five cents to A$7.34.

A coreference resolution system could rule out any possibility of a link between "Softbank's stock" and "News Corp's stock" as long as "Softbank" and "News Corp." are distinct entities.

Such pairs with clearly incompatible modifiers are extremely rare. We usually need some additional information to determine if two properties can apply to the same object. Even the easiest cases of numeric or possessive modifiers can become problematic for an automated approach:

(163) The Washington, D.C.-based company has waited for [about six years]$_1$ to get a license from the Federal Communications Commission to operate a system that would let customers tune into its radio stations anywhere in the country...
If CD Radio gets the license, the company probably will be operating the system in [three years]$_{2,ante=1}$, Margolese said.

A human reader can easily determine that "about six years" and "three years" refer to different periods of time. The same task is very difficult for an automated approach: for example, "about 100 years" and "103 years" are likely to be coreferent. Our system has no knowledge to analyze and compare fuzzy numerals.

Some coreference links are unclear even for human readers:

(164) Vittorio Cecchi Gori is a Rome-based film producer who is turning [his Telemontecarlo and Videomusic television stations]$_1$ into a national network...
Anyone buying the basic package will receive [Cecchi Gori's three stations]$_{2,ante=1}$, Telemontecarlo, Telemontecarlo 2, and Videomusic, as well as BBC World, CNN International, Discovery, MTV Europe, Cartoon Network, and DMX music channel.

One of the anaphor's modifiers, "three", suggests that the antecedents is a group entity consisting of three objects. The antecedent (as proposed by our system) is a set of just two objects, "Telemontecarlo" and "Videomusic". The MUC annotators considered these two markables not coreferent. An alternative analysis is however possible here: "Telemontecarlo" may refer to both "Telemontecarlo" and "Telemontecarlo 2" and the proposed link may, in fact, be correct. An automatic system would have to rely on a sophisticated pro-

cessing scheme here: it would have to identify a numeric modifier ("three") in one markable, build a model for the other markable to enumerate its parts, and assess the compatibility.

The same thematic role can be expressed with syntactically different constructions. We may therefore have to compare pre- and post-modifiers or possessive and non-possessive premodifiers:

(165) "This agreement is the first of many distribution agreements for CNNsi," said [Jim Walton]$_1$, [CNN senior vice president]$_{2,ante=1}$, who heads the new channel.
"We believe that CNNsi will provide our subscribers an in-depth look at the news stories that surround the competition and the athletes," said Denny Wilkinson, [Primestar's senior vice president]$_{3,ante=2}$ of programming and marketing.

A human reader can infer that, as long as "CNN" and "Primestar" are different companies, their presidents are unlikely to be the same person. An automatic system has to determine that "CNN" and "Primestar" play the same role here: for example, "Schumacher's car" and "Ferrari car" can still refer to the same object, although "Schumacher" and "Ferrari" are clearly two distinct entities.

Finally, some modifiers express very different properties that can hardly belong to the same object:

(166) And [in-flight entertainment systems]$_1$ are advancing slowly, as measured by United Airlines' suit against GEC Marconi, which supplied the entertainment gear on United's fleet of 777s.
Live feeds from [commercial digital satellite systems]$_{2,ante=1}$ provided razor-sharp television pictures during a flight.

It is clear for a human reader that "in-flight entertainment systems" and "commercial digital satellite systems" are two distinct objects, but it is very difficult to formalize this intuition and incorporate it into an automated approach.

To summarize, a few (same-head) pairs in our test data have incompatible modifiers. It is however very difficult to automatically establish such incompatibility and thus eliminate the corresponding links.

Some candidate anaphors have modifiers that are very common for discourse new entities. Although they might be compatible with the suggested antecedent's modifiers, the corresponding links are usually incorrect:

(167) If you have a ship that can fire Tomahawk missiles, and fire anti-air missiles, and maybe fire ATACMS (Army Tactical Missiles), [that ship]$_1$ will perform a function that [some other ship]$_{2,ante=1}$ won't have to perform.

Our system has suggested a same-head antecedent for 15 markables with such modifiers. We have presented in Chapter 7 a learning-based algorithm for identifying discourse new descriptions. Our evaluation (Experiments 9 and 10) shows that the system can identify discourse new noun phrase with an F-score of up to 88%, but it nevertheless fails to improve the main coreference resolution algorithm, as long as the MUC scoring scheme is used.

Most spurious links cannot be eliminated by relying just on the modifiers' properties – either at least one markable has no modifiers (62 errors) or they are compatible (58 errors):

(168) Both companies said they expect to use the systems primarily to deliver digital video services to [Latin American subscribers']$_1$ own dishes and to cable company receivers for distribution to [cable subscribers]$_{2,ante=1}$... Turner Broadcasting System Inc., for its part, agreed in July to distribute Cable News Network and three other cable channels to [Latin American subscribers]$_{3,ante=2}$ together with a group called Galaxy Latin America, composed of GM's DirecTV, Venezuela's Cisneros Group of Cos., Brazil's Televisao Abril, and Mexico's MVS Multivision.

"Latin American" and "cable" describe the set of "subscribers" from different perspectives and, therefore, we cannot exclude the possibility of a coreference link between the first and the second markable. The parse tree could be helpful in this case: the pattern "deliver something to A and to B for distribution to C" is a reliable contra-indexing indicator (coreference links between any two of A, B, and C are highly unlikely). The proposed pattern is too specific – an extensive set of such constraints is very hard to compile manually or extract automatically. We can develop more general contra-indexing rules' schemata and try to use data mining techniques to learn such patterns from coreference-annotated data[14].

The third markable, "Latin American subscribers" is identical with the first one, but nevertheless refers to a different entity – "Latin American *cable* subscribers" vs. "Latin American *digital* subscribers". Such distinctions can hardly be captured by a shallow learning-based algorithm.

The markables in (169) have very semantically similar modifiers:

(169) A series of satellite and ground-station links provided [live video]$_1$ from an unmanned reconnaissance aircraft flying over Europe... (5 sentences) A pilot on a bombing run could receive [real-time video]$_{2,ante=1}$ of the

[14]We have investigated a small group of such patterns – the *command* relations advocated by the transformational grammar framework (see Section 4.6). Although they interact with the distribution of coreference links in the documents, these patterns are too general and cannot therefore be used as reliable contra-indexing constraints.

target site, including potential hazards and a suggested route.

Although "live video" and "real-time video" are synonymous, the markables clearly refer to distinct discourse entities. Two factors can help a coreference resolution engine avoid the suggested link. First, the distance between the markables is 5 sentences. "Live video" has been mentioned only once and then forgotten – its salience is very low and it is unlikely to be re-mentioned again. This makes it an implausible antecedent for any (distant) anaphor. Second, "real-time video of the target site"is a post-restrictively modified expression (see Section 4.5) and is therefore likely a discourse new entity. The combination of these two factors makes a coreference link between "live video" and "real-time video" rather improbable.

Some spurious links between same-head expressions could be avoided if our system had a better coverage:

(170) Scheduled for a vote at [the agency's meeting]$_1$ on Thursday, the expected allocation will let the companies transmit video pictures, phone calls, and other data from earth stations to orbiting satellites, and then to customers in Mexico, the Caribbean, Central America, and South America...
A similar plan was set by the International Telecommunications Union at the World Administrative Radio Conference in 1992, and adopted at [the same meeting]$_{2,ante=1}$ in 1995.

"The same meeting" is a likely anaphor, but the suggested antecedent is too far away (14 sentences). An ideal system should relate "the same meeting" to "the World Administrative Radio Conference". Note that it is not the IDENT coreference, as described in the MUC guidelines – these two description clearly refer to distinct events (happening in different years). If our system could account for such coreference phenomena, it would have avoided the spurious antecedent "the agency's meeting" here.

At least one of the markables has no modifier for 62 spurious links proposed by our system. We obviously cannot obtain any relevant information by comparing the (non-existent) modifiers in such cases:

(171) They're touring in support of the just-released album, "Sweet F.A." ([the title]$_1$, [a direct cop]$_{2,ante=1}$ from a 1974 Sweet album, is [English slang]$_{3,ante=1}$ for "sweet nothing"), and they've brought the slinky, layered sound, the glam-rock suggestiveness, the anguish and the angst, the chemical smoke and the stark mood lighting back with them... (13 sentences)
Guitarist Courtney Taylor and synthist Zia McCabe meshed beautifully – complex harmonics shimmering off simple chords – the highlight being

"It's A Fast-Driving Rave-Up With the Dandy Warhols' Sixteen Minutes."
No kidding, that's [the song title]$_{4,ante=1}$.

Our classifier has resolved the fourth markable, "the song title", to the first markable, "the title". The distance between these markables is 14 sentences and the topic has changed several times in between. This example, similar to (171) above, shows how the insufficient coverage of our coreference resolution algorithm (its inability to analyze "that"-pronouns) propagates to decrease the system's precision – when a good candidate is missed, the system suggests some spurious antecedents that would never be even submitted to the classifier otherwise.

The examples we have seen so far clearly suggest that assessing the compatibility of (prenominal) modifiers is a non-trivial problem that cannot be resolved reliably and also cannot help eliminate most spurious same-head links. We should shift to a more global view of the coreference resolution task to get any improvement here.

Our coreference resolution strategy is too local in two senses. First, it only makes pairwise decisions. Each $\{anaphor, candidate\_antecedent\}$ pair is processed almost independently on the coreference chains the system has already constructed[15]. Second, only "basic" NPs are considered to be markables and any information about their embedding NPs is lost. The following examples show how this locality can harm the system's performance.

A local coreference resolution algorithm may merge incompatible markables into one chain:

(172) BC-CD-RADIO-[SHARES]$_1$-BLOOM...
CD Radio stock rose 2 7/8 to 13 5/8 in trading of [400,300 shares]$_{2,ante=1}$, more than quadruple the three-month daily average of [88,700 shares]$_{3,ante=2}$.

Our system has incorrectly resolved the third markable to the second one, although they have incompatible modifiers "400,300" and "88,700". But even if our classifier was informed enough to avoid such links, it would still resolve each of the NPs to the first markable, "SHARES". We cannot eliminate such solutions as long as we have a very local approach: the links to "SHARES" from both the "400,300 shares" and "88,700 shares" candidates are plausible and they can only be avoided by analyzing the resulting chain {"SHARES", "400,300 shares", "88,700 shares"}. Note that a chain-based system would also encounter difficulties here – it would have to decide which markable should be kept in the chain, and which one not. The fact that the system has correctly

[15] We have a group of features to encode some properties of the antecedent's chain (see Section 6.7 for details), but they influence the overall resolution strategy only very indirectly and the system can still merge incompatible markables into the same chain.

ruled out the possibility of a coreference link between two markables is not rewarded by the MUC-7 scorer.

We also have to shift to a deeper analysis of our markables. Our system relies on a parser and an NE-tagger to extract the smallest NP-like units – "basic" noun phrases, names entities, and pronouns (see Section 2.5). The properties of embedding NPs are used very scarcely: we have some syntactic features only applicable to pairs from the same sentence (appositions (Section 4.7), commands (Section 4.6)) and very simplistic features for markables' modifiers ("anaphor/antecedent is pre-/post-modified, Section 4.5). Such approach is very local: we pick a markable out of the context and do not take into account its relations to surrounding NPs. This makes our classifier produce spurious links that seem very plausible on the markable level:

(173) The company also said Marine Corps has begun testing two of [its radars]$_1$ as part of a short-range ballistic missile defense program. That testing could lead to an order for [the radars]$_{2,ante=1}$ that could be worth between $60 million and $70 million.

A coreference link between "its radars" and "the radars" seems plausible if we do not look at the markables' contexts. "Its radars" denote in fact some set of radars and "the radars" – its subset. The annotators have suggested a link between "the radars" and "two". Our system has a very local view of markables, comparing them on the surface level and not attempting any deeper analysis, and therefore it has missed a relation between "two" and "its radars".

To conclude, our system has suggested a spurious same-head antecedent for 182 nominal anaphors. Such links are truly problematic and it is generally assumed in the literature (Poesio and Vieira, 1998) that one should pay attention to (prenominal) modifiers to rule out spurious same-head candidates. We have seen, however, that most suggested links are pairs with compatible or missing modifiers. Pairs with incompatible modifiers are also very difficult, requiring an extensive knowledge base and additional sophisticated inference schemes. We believe that we should pay less attention to the modifiers and look for alternative solutions – improving the overall resolution strategy to have a less local algorithm. This remains an open issue for our future research.

8.3 Discussion

In this chapter we have explored the utility of linguistically motivated features for statistical Coreference Resolution. We have encoded various relevant linguistic factors in 351 feature and evaluated our system on a standard dataset, the MUC-7 corpus, comparing it to the knowledge-poor algorithm proposed by Soon et al. (2001) and a set of naive baselines.

Our Experiment 11 shows that the proposed extension of the feature set results in a consistent improvement in the system's performance. Our model outperforms, on the one hand, classifiers, based on a single knowledge source, and, on the other hand, other algorithms for coreference resolution, evaluated on the same dataset. The learning curves show no signs of convergence, and we believe that substantial improvements can be achieved by adding more training material.

We have also performed a detailed error analysis. We see several major problems with our approach: insufficient data quality, shortages of preprocessing modules, inadequate features and deficiencies in our resolution strategy.

MUC-7 Data and Evaluation Metric. We have used the MUC-7 corpus in our study. It consists of 30 training ("dry-run") and 20 testing ("formal") one-page documents. We have outlined several problems with the theoretic assumptions of the MUC guidelines and the annotation quality.

The definition of IDENT coreference, as advocated by the MUC-7 guidelines, is problematic. van Deemter and Kibble (2001) point out that the MUC annotation scheme fails to separate the coreference relation proper from several other phenomena, such as bound anaphora or predicate nominals. We have seen that the transitivity of the IDENT relation may sometimes lead to counter-intuitive decisions.

The evaluation metric (Vilain et al., 1995) suggested by the MUC guidelines is too biased towards recall. A coreference resolution system is not rewarded directly for avoiding a spurious link. We have seen that even a substantial improvement in the system's precision (by discarding automatically identified discourse new entities, see Chapter 7) does not necessary lead to a better MUC F-score. If we want to use a coreference resolution engine as a preprocessing module for some other engine, for example, an information extraction system, we might want to have a classifier with a high precision level and therefore opt for another scoring scheme, such as the BCUBED metric (Bagga and Baldwin, 1998a).

The corpus is relatively small and does not contain enough material for training (the "formal training" documents provided by MUC-7 are not annotated). Our classifiers show no signs of convergence when we train them on 10, 15, 20, 25, or all the 30 "dry-run" documents (cf. Figure 8.1). We need a larger dataset (for example, the ACE corpus) to make better use of our rich feature set.

The annotation quality is moderate. Deficiencies of manual annotation for the testing corpus inevitably decrease the evaluation score for any system (cf. MUC-7 Inconsistencies in Sections 8.2.1 and 8.2.2). The same problems with the training material make the data noisy and thus potentially deteriorate the performance level of any learning-based approach.

To summarize, a better coreference model can be created by revising the definition of coreference and the scoring scheme and then accurately annotating more training material. As a first step in this direction, we plan to re-train our classifier on an already existing larger corpus (ACE).

Preprocessing modules. We rely on external modules for segmenting MUC documents into sentences[16] (Reynar and Ratnaparkhi, 1997), parsing (Charniak, 2000), NE-tagging (Curran and Clark, 2003b) and determining semantic properties of our markables (Miller, 1990). The first three modules are fully automatic corpus-based NLP systems. The WordNet ontology is a large manually created resource.

All the modules have some shortages that may decrease the performance of our system. We have seen, for example, that appositive-coordinate constructions are intrinsically difficult for parsing. This results in incorrect markables and spurious or missing links. A possible remedy would be creating a family of mini-parsers, specifically trained to analyze problematic constructions relevant for coreference resolution.

External modules help extract different kinds of information from raw text data: parse trees, NE-tags, or semantic labels for specific words. We have however to design additional rules, transforming this knowledge into the information required by our system, extracting relevant bits of the modules' output and encoding them as features. For example, our system misses markables that are not full NPs, but prenominal modifiers, even if they are labelled correctly by the parser. The WordNet ontology does not contain exactly the information needed for coreference resolution: although commonly used WordNet similarity measures correlate with coreference, the corresponding features do not affect the classifier. We have to design more elaborate techniques to obtain measures of semantic compatibility from the WordNet data.

To summarize, our engine relies on a number of off-the-shelf preprocessing systems These modules are error prone and their output is not exactly what we need. We have to improve the interaction with our preprocessing modules – adjust the external resources to cover specific problems relevant for our task (e.g., train a mini-parser for appositive-coordinate constructions or an NE-tagger for titles) and design better strategies for transforming the output of (general-purpose) modules into the information (features and markables) needed for our system.

Features. Our classifier relies on 351 feature (1096 boolean/continuous). Not all of them are equally important. We were not able to perform any feature selection within this study and therefore our feature set is highly redundant. Ng and Cardie (2002c) have shown that (manual) feature selection

[16] We have not encountered any errors directly caused by incorrect sentence segmentation.

can significantly improve the performance level of a linguistically motivated coreference resolution algorithm.

Some phenomena are covered by too many features simultaneously. Most of our name-matching and salience features are produced by enumerating and combining possible values for a set of parameters. This results in a pool of highly inter-correlated features. Even though each feature brings some important bit of information, the whole set has a too high degree of redundancy for machine learning. Future experiments within this framework should address the problem of feature selection — reducing the number of features to get a better classifier.

Some phenomena are covered by our feature set, but the corresponding features are almost ignored by the classifier. For example, we have features to account for abbreviations, but neither C4.5, nor SVM^{light}make any use of them. Our training data does not contain enough abbreviations to learn any reliable patterns. We have to increase the training corpus to get better results.

Finally, some phenomena are still not covered by our feature set. For example, we do not account for NP modifiers and their compatibility. On the one hand, there is always room for improvement: even a system with millions of features can always be augmented with some new information. On the other hand, obtaining values for more sophisticated features is a very difficult task: we need additional external resources and they are likely to introduce errors. We believe that our system already has a lot of encoded information and therefore we have to improve the algorithm itself rather than introduce more knowledge. This view is supported by our learning curves: they show no signs of convergence, suggesting that we still can get better results with the same feature set.

To summarize, our system relies on 351 features (1096 boolean/continuous) covering linguistic properties of markables and markable pairs relevant for coreference resolution. Some phenomena are over-represented in our feature set, decreasing the classifier's performance. A few phenomena, on the contrary, are not covered well enough. We plan to investigate feature selection and ensemble learning with different feature splits to make better use of our features.

Resolution strategy. Our system a very simple resolution scheme: candidate antecedents for each anaphor are proposed to the classifier one-by-one from right to left until a positive instance is found. This scheme was suggested by Soon et al. (2001) and then followed by most studies on coreference. The strategy is very local and does not take into account any other markables, when establishing a link between an anaphor and a candidate antecedent.

We have seen above that it may lead to error propagation. If our classifier has suggested a spurious antecedent for some markable at an early processing

stage (precision error), it will never see any truly positive testing instances and will be unable to resolve the anaphor (recall error). If the classifier has missed the correct antecedent (recall error), it starts processing too distant markables and is likely to suggest some spurious markable (precision error). This can be avoided by adjusting the algorithm's search strategy and making it less local.

Our system sometimes merges several chains into one – it finds pairs of markables (belonging to different chains in the manually annotated data) that seem to be coreferent and links them. The properties of other markables from the affected chains are completely ignored. This problem also could be avoided by shifting to a more global resolution strategy, operating on chains instead of markables. Theoretic studies of coreference usually have a global view, talking, for example, about "discourse entities". Practical approaches, however, almost never go beyond the markable level. The only algorithm operating directly on chains has been advocated by Luo et al. (2004). For an overview of studies incorporating some of chains' properties into their feature sets see Section 6.7.

Finally, our system processes all markables in a uniform way. We have seen, however, that various types of anaphors may require very different linguistic knowledge for their resolution. Our uniform strategy, for example, accounts for almost all precision errors in pronominal anaphora resolution. We have to identify several homogeneous sub-tasks to learn mini-classifiers and incorporate them into the main coreference resolution engine.

To summarize, we have proposed a learning-based coreference resolution system that relies on a rich feature set, created by examining various predictions of linguistic theories. The system shows significant improvement over the baseline and outperforms all the other algorithms evaluated on the same dataset. The evaluation results, in particular our learning curves, show that our study makes a good basis for further experimentation. Our approach can be further developed by elaborating on its resolution part — the main focus of this thesis lies on the utility of linguistic knowledge and not on improving the processing strategies. The detailed error analysis hints at possible directions for future work.

8.4 Summary

In this chapter we have combined the knowledge types investigated so far to build a linguistically-motivated coreference resolution engine. Our system relies on a rich set of name-matching, syntactic, semantic, and salience features.

In Experiment 11 (Section 8.1) we have evaluated the system's performance with a baseline feature set (Soon et al., 2001) and with our extended feature set for a variety of machine learners. We have seen that our linguistic features bring consistent improvement over the basic setting. The system's performance with the SVM^{light} learner is the best result reported for the MUC-7 data

in the literature. The learning curves show no sign of convergence for the extended feature set, suggesting that we can expect further improvements with additional training material.

We have performed a detailed error analysis (Section 8.2). It has revealed several major flaws of our approach: moderate data quality, low interaction with preprocessing modules, high redundancy of the feature set and too local resolution strategy. We have discussed these problems and highlighted possible directions for future work in Section 8.3.

Chapter 9

Conclusion

In this final chapter we summarize the central findings of the thesis, discuss the main issues, raised by our work, and highlight possible directions for future work.

9.1 Main Findings

This study was devoted to bridging the gap between theoretical studies on coreference resolution and state-of-the-art statistical algorithms for the task. We investigated the possibility of incorporating different kinds of linguistic knowledge, suggested by theoretical research, into a machine learning framework. Our experiments resulted in a number of findings concerning, on the one hand, the empirical appropriateness of the investigated theoretical claims, and, on the other hand, their usability for a data-driven resolution algorithm.

First, we assessed the validity of the theoretical predictions, suggested in the literature, by computing distributions of the corresponding features on the MUC-7 corpus and investigating their interaction with coreference. These data supported the hypothesis that complex linguistic parameters of NPs and NP pairs, mostly ignored by state-of-the-art coreference resolution systems, affect the distribution of anaphoric links in the document and constitute, therefore, a valuable source of information.

At the same time, the evaluation revealed several problematic issues with the investigated linguistic factors. Most theoretical predictions addressed in our study are based on small sets of precompiled examples and have, therefore, only restricted coverage. The corpus-based analysis, presented in this thesis, could help discover novel linguistic patterns, relevant for coreference resolution, or refine the existing ones, resulting in more accurate claims.

Second, we investigated the utility of complex linguistic structures, dependent on the assumptions of theoretical research, for statistical coreference resolution. We encoded these structures as features to build a linguistically motivated data-driven coreference resolution system. This framework allowed us to combine the robustness and flexibility of a learning-based approach with the linguistic information, suggested by theoretical studies on coreference. We evaluated our model on a standard dataset (the MUC-7 corpus) with a traditional learning set-up (Soon et al., 2001).

Our evaluation experiments confirmed the hypothesis that generalizations of linguistic theories can be effectively incorporated into a machine learning framework: theoretical claims, encoded as features, may indeed improve the performance level of a statistical coreference resolution system. This finding is important for application-oriented studies in the field: most existing coreference resolution systems show similar performance figures, failing to resolve essentially the same set of "difficult anaphors" (Cristea et al., 2002). Our model is potentially capable of tackling exactly those more difficult links, by relying on more sophisticated linguistic knowledge.

Third, our research created a basis for further experimentation on statistical coreference resolution. The learning curves suggested that our simple baseline system almost achieved the upper-bound for its performance already with a very small amount of training material and could hardly be improved any further. The learning curves for the linguistically informed model, on the contrary, confirmed the potential of our model — we can further improve our coreference resolution system by annotating more training material or by elaborating on the processing scheme to use the existing data more intelligently.

Fourth, we performed and reported a detailed error analysis. It provided valuable information for applied studies on coreference, presenting a distribution of problematic cases. The error analysis raised a number of issues with the theoretical assumptions of the MUC-7 algorithm and the resolution scheme, adopted by our system. We see this part of our study as a first step toward future research: it allowed us to formulate several mostly unaddressed problems in the field of statistical coreference resolution.

Finally, we proposed a model achieving the best performance level reported for the MUC-7 dataset. The main focus of our thesis was on the utility of complex theoretical predictions for statistical coreference resolution. Improving the underlying resolution strategy was outside the scope of our study: we relied on an existing dataset and followed a traditional processing scheme. These restrictions allowed us to directly assess the impact of linguistic knowledge on the overall performance, but, at the same time, made it infeasible to build the best possible coreference resolution system. Our model, nevertheless, achieved significant improvement over resolution algorithms, oriented on a single information source, and over the state-of-the-art. To our knowledge, the performance of our system with the SVM^{light} learner (F-score of 65.4%) is

the best result on the MUC-7 data reported so far in the literature.

9.2 Future Work

Our experiments raise a number of issues for both theoretical and application-oriented research on coreference. In this section we highlight directions for future research, suggested by this thesis.

Linguistic knowledge. Our experiments generally confirm the finding that theoretical claims on coreference may bring valuable information for an application-oriented approach. They reveal, at the same time, a coverage problem: most claims are based on small sets of manually crafted examples and can not, therefore, fully account for real-world texts. For example, the contra-indexing command constraints, as originally formulated within the generative grammar framework, are not reliable indicators against coreference. We had to modify the original definition in order to make the command relations useful for our algorithm, accounting for specific constructions and most common parsing errors.

The error analysis reveals two problematic areas, where the existing methods are virtually unable to distinguish between pairs of coreferring vs. non-coreferring markables: intra-sentential coreference (apart from appositions and copulas) and nominal anaphora. In both cases, linguistic theories identify only a small amount of relevant patterns.

We see two possible extensions here. One can use the collected corpus data to manually inspect and adjust relevant theoretical predictions. Our "modified" features are just the first step in this direction.

A more challenging task involves using data mining techniques to automatically adjust the claims. We can try to extract patterns of coreference, using the existing predictions as a starting point (for example, as seed items). This could help us to overcome the coverage problem. Moreover, we could quickly re-adjust the set of constraints for new domains and corpora. Such an approach would be beneficial for both theoretical and applied studies on coreference. We believe that automatic acquisition of indicators for and against coreference is an interesting problem on the edge of theory and practice.

Annotating Coreference. Our study suggests a number of extensions, concerning the quantity and the quality of the annotated material. First, the learning curves for our linguistically informed system show no signs of convergence. Second, the error analysis reveals several inconsistencies of the MUC-7 corpus, affecting the system's performance. This makes us believe that we can further improve our coreference resolution engine by using more data and revising the annotation guidelines.

The possible extensions mainly concern the view of coreference, adopted by the MUC committee. We believe that the revised guidelines should provide a more formal and uniform definition, separating coreference proper from other related phenomena. The corpus should ideally represent a single domain (or the same set of domains for the testing and the training parts) and contain pure texts, without any semi-structured auxiliary parts. This would make the annotated relation more homogeneous and therefore better suitable for machine learning. It is worth noting that our suggestions, motivated by corpus-based evaluation experiments, are in accordance with the criticism for the MUC-7 approach from a theoretical perspective (van Deemter and Kibble, 2001).

Designing new guidelines and annotating an extensive amount of data is an extremely hard and time-consuming task. We plan, therefore, to use another existing corpus, the ACE dataset (NIST, 2003), to get additional learning material for our algorithm[1].

Architecture. Our error analysis shows that inaccuracies of various external modules significantly decrease the system's performance, being, for example, one of the main sources of recall errors. An important open issue for future research concerns possible internal structure of any coreference resolution system and, more generally, of any complex linguistic engine.

Large and complex NLP engines rely on several error-prone sub-systems. Such modules are often used as black boxes. For example, one may take an off-the-shelf parser and rely on its output for extracting syntactic knowledge: the main system does not send any feedback to the parser and does not attempt to detect and correct parsing mistakes, affecting the ultimate performance. An NLP engine with such an architecture will probably aggregate errors propagated from its sub-systems.

We believe that it is a challenging research problem to develop a probabilistic model that intelligently combines various modules. Such model could help neutralize errors, introduced by different subsystems (a simple solution would be employing a committee voting-like scenario). It could also help optimize different modules w.r.t the ultimate task.

We have explored a few steps in this direction. For example, we compare NP boundaries, suggested by the parser and by the NE-tagger, and discard mutually incompatible solutions to create a pool of markables. Our algorithm for detecting discourse new entities helps identify and discard some erroneous noun phrases. These are, however, rather preliminary and unsystematic steps. Future research should concentrate on developing a uniform framework for multi-module coreference resolution systems.

[1]These data were not publicly available at the beginning of our experiments and are still distributed under restrictions.

Machine Learning. Our experimental results show that the classifiers have not reached the upper bound for their performance and still have a potential for improvements. We have already discussed the possibility of learning a better prediction function by adding more training material. A challenging problem for future research would be to investigate the strategies for using the already existing data more intelligently.

Further experiments should incorporate ensemble learning and automatic feature selection into the framework. Some steps in this direction have been suggested, for example, by Strube et al. (2002a) and Ng and Cardie (2002c). Strube et al. (2002a) have obtained mostly negative results using co-training on German coreference data. Ng and Cardie (2002c) have investigated manual feature selection and reported improvements of their system's performance on the MUC data. Our approach, however, is different from these systems — we rely on a very rich feature set, enabling more elaborated experiments on feature selection and feature splits for ensemble learning.

Resolution Strategy. The learning curves for our final experiments show no signs of convergence, suggesting that better performance figures can still be achieved simply by improving the machine learning component of our algorithm. Future studies should, however, go further and elaborate on the whole framework.

We deliberately restricted the scope of our thesis to follow the commonly adopted resolution strategy of Soon et al. (2001). Our linguistically informed system yielded a reliable performance even with such a simple processing scheme. We can, however, improve the resolution strategy in several respects.

One possible extension concerns more linguistically motivated resolution strategies. Theoretical studies both in linguistics (Gundel et al., 1993) and psychology (Garrod et al., 1994) clearly suggest different processing mechanisms for various types of anaphors. We could, therefore, split the task of full-scale coreference resolution into sub-problems, propose separate solutions and then intelligently combine them in a complex model. The exact definition of sub-problems should be motivated by the linguistic theory. Such an approach could rely, for example, on distinct sub-systems for intra- vs. intersentential coreference or for pronouns vs. proper names vs. nominal anaphora. Each sub-system would rely on a separate feature set and employ a specific sample selection strategy or even have an own distinct resolution algorithm. We have explored a step in this direction in (Uryupina, 2004).

Another, more radical extension, would be to completely revise the resolution strategy advocated by Soon et al. (2001). Our evaluation experiments identified a number of problems with this setting. In particular, it is very local: the classifier operates on pairs of markables and essentially ignores all the other information. Theoretical studies on coreference, on the contrary, mostly oper-

ate on the level of "discourse entities", or coreference chains. Further research in this area has to investigate the possibility of upgrading the whole framework and bringing it to the chains level. This would allow, on the one hand, to improve the performance by making the resolution strategy less local and, on the other hand, to more accurately test the predictions of various discourse theories. This is a very complex problem and we are aware of only two algorithms proposing full-scale coreference models on the chains level (Cardie and Wagstaff, 1999; Luo et al., 2004). It must also be kept in mind that the commonly adopted approach of Soon et al. (2001) recasts coreference resolution as a simple pattern recognition problem, addressed in a variety of statistical learning frameworks, and allows therefore to test numerous machine learners. It is not clear whether a comparable performance level can be achieved with state-of-the-art AI methods for sequence modeling.

To summarize, this thesis investigates the possibility of incorporating wide variety of data from different levels of linguistic description into a statistical model of coreference. We assess the accuracy of theoretical claims for real-world data by encoding them as features and examining the corresponding distributions and their interaction with coreference. This part of the thesis provides feedback for theoretical studies on the problem. We use our name-matching, syntactic, semantic and discourse features to build a linguistically informed statistical coreference resolution engine. This allows us to combine the robustness and flexibility of a learning-based approach with the linguistic information, investigated in various theoretical studies. Our evaluation experiments confirm the hypothesis that a statistical coreference resolution algorithm may benefit from such a rich feature set: our linguistically informed system outperforms all the four single-source classifiers and yields the best result (F-score of 65.4%) reported for the MUC data in the literature. Our thesis provides a basis for further experiments on statistical coreference resolution: the learning curves show no signs of convergence, suggesting that even better results can be achieved by elaborating on the machine learning component and refining the resolution strategy.

Appendix A

List of Features

Below we list all the features, discussed in this thesis, with the references to the sections where they are introduced. The features are used to describe pairs of markables, (M_i, M_j), where M_i is a (candidate) pronoun and M_j is a (candidate) antecedent.

Feature	Range	Section
Features of Soon et al. (2001)		
DIST(M_i, M_j)	continuous	2.3
I_PRONOUN(M_i, M_j)	0,1	2.3
J_PRONOUN(M_i, M_j)	0,1	2.3
STR_MATCH(M_i, M_j)	0,1	2.3
DEF_NP(M_i, M_j)	0,1	2.3
DEM_NP(M_i, M_j)	0,1	2.3
NUMBER(M_i, M_j)	0,1	2.3
SEMCLASS(M_i, M_j)	0,1,?	2.3
GENDER(M_i, M_j)	0,1,?	2.3
PROP_NAME(M_i, M_j)	0,1	2.3
ALIAS(M_i, M_j)	0,1	2.3
APPOSITIVE(M_i, M_j)	0,1	2.3
Name-matching features		
lower_case(M_i)	0,1	3.4
cap_words(M_i)	0,1	3.4
upper_case(M_i)	0,1	3.4
digits(M_i)	0,1	3.4
alphas(M_i)	0,1	3.4
lower_case_h(M_i)	0,1	3.4
cap_words_h(M_i)	0,1	3.4

upper_case_h(M_i)	0,1	3.4
digits_h(M_i)	0,1	3.4
alphas_h(M_i)	0,1	3.4
rarest(M_i)	continuous	3.4
length_s(M_i)	continuous	3.4
length_w(M_i)	continuous	3.4
lower_case(M_j)	0,1	3.4
cap_words(M_j)	0,1	3.4
upper_case(M_j)	0,1	3.4
digits(M_j)	0,1	3.4
alphas(M_j)	0,1	3.4
lower_case_h(M_j)	0,1	3.4
cap_words_h(M_j)	0,1	3.4
upper_case_h(M_j)	0,1	3.4
digits_h(M_j)	0,1	3.4
alphas_h(M_j)	0,1	3.4
rarest(M_j)	continuous	3.4
length_s(M_j)	continuous	3.4
length_w(M_j)	continuous	3.4
MED_w(M_i,M_j)	continuous	3.4
MED_w(no_det(M_i,M_j))	continuous	3.4
MED_s(M_i,M_j)	continuous	3.4
MED_s(no_det(M_i,M_j))	continuous	3.4
MED_s(head(M_i,M_j)	continuous	3.4
MED_w(no_case(M_i,M_j))	continuous	3.4
MED_w(no_case(no_det(M_i,M_j)))	continuous	3.4
MED_s(no_case(M_i,M_j))	continuous	3.4
MED_s(no_case(no_det(M_i,M_j)))	continuous	3.4
MED_s(head(no_case(M_i,M_j)))	continuous	3.4
MED_w(no_punct(M_i,M_j))	continuous	3.4
MED_w(no_punct(no_det(M_i,M_j)))	continuous	3.4
MED_s(no_punct(M_i,M_j))	continuous	3.4
MED_s(no_punct(no_det(M_i,M_j)))	continuous	3.4
MED_w(no_case(no_punct(M_i,M_j)))	continuous	3.4
MED_w(no_case(no_punct(no_det(M_i,M_j))))	continuous	3.4
MED_s(no_case(no_punct(M_i,M_j)))	continuous	3.4
MED_s(no_case(no_punct(no_det(M_i,M_j))))	continuous	3.4
MED_w_anaph(M_i,M_j)	continuous	3.4
MED_w_anaph(no_det(M_i,M_j))	continuous	3.4
MED_s_anaph(M_i,M_j)	continuous	3.4
MED_s_anaph(no_det(M_i,M_j))	continuous	3.4
MED_s_anaph(head(M_i,M_j))	continuous	3.4

MED_w_anaph(no_case(M_i,M_j))	continuous	3.4
MED_w_anaph(no_case(no_det(M_i,M_j)))	continuous	3.4
MED_s_anaph(no_case(M_i,M_j))	continuous	3.4
MED_s_anaph(no_case(no_det(M_i,M_j)))	continuous	3.4
MED_s_anaph(head(no_case(M_i,M_j)))	continuous	3.4
MED_w_anaph(no_punct(M_i,M_j))	continuous	3.4
MED_w_anaph(no_punct(no_det(M_i,M_j)))	continuous	3.4
MED_s_anaph(no_punct(M_i,M_j))	continuous	3.4
MED_s_anaph(no_punct(no_det(M_i,M_j)))	continuous	3.4
MED_w_anaph(no_case(no_punct(M_i,M_j)))	continuous	3.4
MED_w_anaph(no_case(no_punct(no_det(M_i,M_j))))	continuous	3.4
MED_s_anaph(no_case(no_punct(M_i,M_j)))	continuous	3.4
MED_s_anaph(no_case(no_punct(no_det(M_i,M_j))))	continuous	3.4
MED_w_ante(M_i,M_j)	continuous	3.4
MED_w_ante(no_det(M_i,M_j))	continuous	3.4
MED_s_ante(M_i,M_j)	continuous	3.4
MED_s_ante(no_det(M_i,M_j))	continuous	3.4
MED_s_ante(head(M_i,M_j))	continuous	3.4
MED_w_ante(no_case(M_i,M_j))	continuous	3.4
MED_w_ante(no_case(no_det(M_i,M_j)))	continuous	3.4
MED_s_ante(no_case(M_i,M_j))	continuous	3.4
MED_s_ante(no_case(no_det(M_i,M_j)))	continuous	3.4
MED_s_ante(head(no_case(M_i,M_j)))	continuous	3.4
MED_w_ante(no_punct(M_i,M_j))	continuous	3.4
MED_w_ante(no_punct(no_det(M_i,M_j)))	continuous	3.4
MED_s_ante(no_punct(M_i,M_j))	continuous	3.4
MED_s_ante(no_punct(no_det(M_i,M_j)))	continuous	3.4
MED_w_ante(no_case(no_punct(M_i,M_j)))	continuous	3.4
MED_w_ante(no_case(no_punct(no_det(M_i,M_j))))	continuous	3.4
MED_s_ante(no_case(no_punct(M_i,M_j)))	continuous	3.4
MED_s_ante(no_case(no_punct(no_det(M_i,M_j))))	continuous	3.4
abbrev1(M_i,M_j)	0,1	3.4
abbrev1(no_case(M_i,M_j))	0,1	3.4
abbrev2(M_i,M_j)	0,1	3.4
abbrev2(no_case(M_i,M_j))	0,1	3.4
abbrev3(M_i,M_j)	0,1	3.4
abbrev3(no_case(M_i,M_j))	0,1	3.4
abbrev3(no_punct(M_i,M_j))	0,1	3.4
abbrev3(no_case(no_punct(M_i,M_j)))	0,1	3.4
abbrev4(M_i,M_j)	0,1	3.4
abbrev4(no_case(M_i,M_j))	0,1	3.4
abbrev4(no_punct(M_i,M_j))	0,1	3.4

abbrev4(no_case(no_punct(M_i,M_j)))	0,1	3.4
exact_match(head(M_i,M_j))	0,1	3.4
exact_match(head(no_case(M_i,M_j)))	0,1	3.4
exact_match(M_i,M_j)	0,1	3.4
exact_match(no_case(M_i,M_j))	0,1	3.4
exact_match(no_punct(M_i,M_j))	0,1	3.4
exact_match(no_case(no_punct(M_i,M_j)))	0,1	3.4
exact_match(no_det(M_i,M_j))	0,1	3.4
exact_match(no_case(no_det(M_i,M_j)))	0,1	3.4
exact_match(no_punct(no_det(M_i,M_j)))	0,1	3.4
exact_match(no_case(no_punct(no_det(M_i,M_j))))	0,1	3.4
exact_match(rarest(M_i,M_j))	0,1	3.4
exact_match(rarest(no_case(M_i,M_j)))	0,1	3.4
rarest+contain(M_i,M_j)	0,1	3.4
rarest+contain(no_case(M_i,M_j))	0,1	3.4
rarest+contain(M_j,M_i)	0,1	3.4
rarest+contain(no_case(M_j,M_i))	0,1	3.4
exact_match(first(M_i,M_j))	0,1	3.4
exact_match(first(no_det(M_i,M_j)))	0,1	3.4
exact_match(firstnotitle(no_det(M_i,M_j)))	0,1	3.4
exact_match(first(no_case(M_i,M_j))	0,1	3.4
exact_match(first(no_case(no_det(M_i,M_j)))	0,1	3.4
exact_match(firstnotitle(no_case(no_det(M_i,M_j))))	0,1	3.4
exact_match(last(M_i,M_j))	0,1	3.4
exact_match(last(no_case(M_i,M_j))	0,1	3.4
matched_part_w(M_i, M_j)	continuous	3.4
matched_part_w(no_case(M_i, M_j))	continuous	3.4
matched_part_w(no_punct(M_i, M_j))	continuous	3.4
matched_part_w(no_det(M_i, M_j))	continuous	3.4
matched_part_w(no_case(no_det(M_i, M_j)))	continuous	3.4
matched_part_w(no_case(no_punct(M_i, M_j)))	continuous	3.4
matched_part_w(no_det(no_punct(M_i, M_j)))	continuous	3.4
matched_part_w(no_case(no_det(no_punct(M_i, M_j)))	continuous	3.4
matched_part_s(M_i, M_j)	continuous	3.4
matched_part_s(no_case(M_i, M_j))	continuous	3.4
matched_part_s(no_punct(M_i, M_j))	continuous	3.4
matched_part_s(no_det(M_i, M_j))	continuous	3.4
matched_part_s(no_case(no_det(M_i, M_j)))	continuous	3.4
matched_part_s(no_case(no_punct(M_i, M_j)))	continuous	3.4
matched_part_s(no_det(no_punct(M_i, M_j)))	continuous	3.4
matched_part_s(no_case(no_det(no_punct(M_i, M_j)))	continuous	3.4

Syntactic features		
type_of_markable(M_i)	nominal	4.2
type_of_pronoun(M_i)	nominal	4.2
type_of_definite(M_i)	nominal	4.2
determiner(M_i)	nominal	4.3
det_ana_type(M_i)	nominal	4.3
det_ante_type(M_i)	nominal	4.3
head_anaphoric(M_i)	0,1	4.4
head_nonanaphoric(M_i)	0,1	4.4
head_antecedent(M_i)	0,1	4.4
head_nonantecedent(M_i)	0,1	4.4
coordination(M_i)	0,1	4.5
premodified(M_i)	0,1	4.5
postmodified(M_i)	0,1	4.5
postrestrictive(M_i)	0,1	4.5
grammatical_role(M_i)	nominal	4.8
subject(M_i)	0,1	4.8, 6.6
sentence_subject(M_i)	0,1	4.8, 6.6
minimal_depth_subject(M_i)	0,1	4.8
number(M_i)	nominal	4.9
person(M_i)	nominal	4.9
type_of_markable(M_j)	nominal	4.2
type_of_pronoun(M_j)	nominal	4.2
type_of_definite(M_j)	nominal	4.2
determiner(M_j)	nominal	4.3
det_ana_type(M_j)	nominal	4.3
det_ante_type(M_j)	nominal	4.3
head_anaphoric(M_j)	0,1	4.4
head_nonanaphoric(M_j)	0,1	4.4
head_antecedent(M_j)	0,1	4.4
head_nonantecedent(M_j)	0,1	4.4
coordination(M_j)	0,1	4.5
premodified(M_j)	0,1	4.5
postmodified(M_j)	0,1	4.5
postrestrictive(M_j)	0,1	4.5
grammatical_role(M_j)	nominal	4.8
subject(M_j)	0,1	4.8, 6.6
sentence_subject(M_j)	0,1	4.8, 6.6
minimal_depth_subject(M_j)	0,1	4.8
number(M_j)	nominal	4.9
person(M_j)	nominal	4.9
ccommand(M_i, M_j)	0,1	4.6

scommand(M_i, M_j)	0,1	4.6
rcommand(M_i, M_j)	0,1	4.6
ccommand_modified(M_i, M_j)	0,1	4.6
scommand_modified(M_i, M_j)	0,1	4.6
rcommand_modified(M_i, M_j)	0,1	4.6
apposition_basic(M_i, M_j)	0,1	4.6
apposition(M_i, M_j)	0,1	4.7
copula_present(M_i, M_j)	0,1	4.7
copula_all(M_i, M_j)	0,1	4.7
copula_all_notmodal(M_i, M_j)	0,1	4.7
copula(M_i, M_j)	0,1	4.7
same_number(M_i, M_j)	0,1	4.9
same_person(M_i, M_j)	0,1	4.9
same_person_quoted(M_i, M_j)	0,1	4.9
synt_agree(M_i, M_j)	0,1	4.9
synt_agree_quoted(M_i, M_j)	0,1	4.9
parallel(M_i, M_j)	0,1	4.8
parallel_pronoun(M_i, M_j)	0,1	4.8
Semantic features		
semclass_ne(M_i)	nominal	5.2
semclass_soon(M_i)	nominal	5.2
gender(M_i)	nominal	5.2
semclass_ne(M_j)	nominal	5.2
semclass_soon(M_j)	nominal	5.2
gender(M_j)	nominal	5.2
same_semclass_wordnet(M_i, M_j)	0,1	5.2
same_semclass_ne(M_i, M_j)	0,1	5.2
same_semclass_soon(M_i, M_j)	0,1	5.2
same_gender(M_i, M_j)	0,1	5.2
compatible_semclass_wordnet(M_i, M_j)	0,1	5.2
compatible_semclass_ne(M_i, M_j)	0,1	5.2
compatible_semclass_soon(M_i, M_j)	0,1	5.2
compatible_gender(M_i, M_j)	0,1	5.2
leacock_firstsense(M_i, M_j)	continuous	5.3
leacock_max(M_i, M_j)	continuous	5.3
resnik_firstsense(M_i, M_j)	continuous	5.3
resnik_max(M_i, M_j)	continuous	5.3
lin_firstsense(M_i, M_j)	continuous	5.3
lin_max(M_i, M_j)	continuous	5.3
jiang_firstsense(M_i, M_j)	continuous	5.3
jiang_max(M_i, M_j)	continuous	5.3
lso(M_i, M_j)	nominal	5.4

lso_d(M_i, M_j)	nominal	5.4
ana_lso_d(M_i, M_j)	nominal	5.4
ante_lso_d(M_i, M_j)	nominal	5.4
{ana_lso_d, ante_lso_d}(M_i, M_j)	nominal	5.4
{ana_lso_d, ante_lso_d, lso}(M_i, M_j)	nominal	5.4
{ana_lso_d, ante_lso_d, lso_d}(M_i, M_j)	nominal	5.4
Discourse features		
section_tag(M_i)	nominal	6.2
paragraph_number_bin(M_i)	1...10	6.2
sentence_number_bin(M_i)	1...10	6.2
paragraph_rank_bin(M_i)	1...10	6.2
sentence_rank_bin(M_i)	1...10	6.2
embedded(M_i)	0,1	6.3
cb(M_i)	0,1	6.5
first_in_sentence(M_i)	0,1	6.2
first_in_paragraph(M_i)	0,1	6.2
section_tag(M_j)	nominal	6.2
pararaph_number_bin(M_j)	1...10	6.2
sentence_number_bin(M_j)	1...10	6.2
paragraph_rank_bin(M_j)	1...10	6.2
sentence_rank_bin(M_j)	1...10	6.2
embedded(M_j)	0,1	6.3
cb(M_j)	0,1	6.5
first_in_sentence(M_j)	0,1	6.2
first_in_paragraph(M_j)	0,1	6.2
ante_discourse_old(M_j)	0,1	6.7
chains_size(M_j)	continuous	6.7
chains_size_same_paragraph(M_j)	continuous	6.7
ante_ante_marktype(M_j)	nominal	6.7
ante_ante_netag(M_j)	nominal	6.7
ante_ante_sent_subject(M_j)	0,1	6.7
ante_ante_subject(M_j)	0,1	6.7
ante_ante_sfirst(M_j)	0,1	6.7
ante_ante_pfirst(M_j)	0,1	6.7
ante_ante_cb(M_j)	0,1	6.7
paragraph_distance(M_i,M_j)	continuous	6.4
sentence_distance(M_i,M_j)	continuous	6.4
markable_distance(M_i,M_j)	continuous	6.4
same_sentence(M_i,M_j)	0,1	6.4
same_paragraph(M_i,M_j)	0,1	6.4
subject_prev_agree(M_i,M_j)	0,1	6.6
subject_same_agree(M_i,M_j)	0,1	6.6

subject_closest_agree(M_i,M_j)	0,1	6.6
ssubject_prev_agree(M_i,M_j)	0,1	6.6
ssubject_same_agree(M_i,M_j)	0,1	6.6
ssubject_closest_agree(M_i,M_j)	0,1	6.6
closest_closest_agree(M_i,M_j)	0,1	6.6
closest_prev_agree(M_i,M_j)	0,1	6.6
closest_same_agree(M_i,M_j)	0,1	6.6
sfirst_prev_agree(M_i,M_j)	0,1	6.6
sfirst_same_agree(M_i,M_j)	0,1	6.6
sfirst_closest_agree(M_i,M_j)	0,1	6.6
pfirst_prev_agree(M_i,M_j)	0,1	6.6
pfirst_same_agree(M_i,M_j)	0,1	6.6
pfirst_closest_agree(M_i,M_j)	0,1	6.6
cb_closest_agree(M_i,M_j)	0,1	6.6
cb_prev_agree(M_i,M_j)	0,1	6.6
cb_same_agree(M_i,M_j)	0,1	6.6
subject_prev(M_i,M_j)	0,1	6.6
subject_same(M_i,M_j)	0,1	6.6
closest_prev(M_i,M_j)	0,1	6.6
closest_same(M_i,M_j)	0,1	6.6
sfirst_prev(M_i,M_j)	0,1	6.6
sfirst_same(M_i,M_j)	0,1	6.6
pfirst_prev(M_i,M_j)	0,1	6.6
pfirst_same(M_i,M_j)	0,1	6.6
ssubject_prev(M_i,M_j)	0,1	6.6
ssubject_same(M_i,M_j)	0,1	6.6
cb_prev(M_i,M_j)	0,1	6.6
cb_same(M_i,M_j)	0,1	6.6
proana_subject_prev_agree(M_i,M_j)	0,1	6.6
proana_subject_same_agree(M_i,M_j)	0,1	6.6
proana_subject_closest_agree(M_i,M_j)	0,1	6.6
proana_ssubject_prev_agree(M_i,M_j)	0,1	6.6
proana_ssubject_same_agree(M_i,M_j)	0,1	6.6
proana_ssubject_closest_agree(M_i,M_j)	0,1	6.6
proana_closest_closest_agree(M_i,M_j)	0,1	6.6
proana_closest_prev_agree(M_i,M_j)	0,1	6.6
proana_closest_same_agree(M_i,M_j)	0,1	6.6
proana_sfirst_prev_agree(M_i,M_j)	0,1	6.6
proana_sfirst_same_agree(M_i,M_j)	0,1	6.6
proana_sfirst_closest_agree(M_i,M_j)	0,1	6.6
proana_pfirst_prev_agree(M_i,M_j)	0,1	6.6
proana_pfirst_same_agree(M_i,M_j)	0,1	6.6

proana_pfirst_closest_agree(M_i,M_j)	0,1	6.6
proana_cb_closest_agree(M_i,M_j)	0,1	6.6
proana_cb_prev_agree(M_i,M_j)	0,1	6.6
proana_cb_same_agree(M_i,M_j)	0,1	6.6
proana_subject_prev(M_i,M_j)	0,1	6.6
proana_subject_same(M_i,M_j)	0,1	6.6
proana_closest_prev(M_i,M_j)	0,1	6.6
proana_closest_same(M_i,M_j)	0,1	6.6
proana_sfirst_prev(M_i,M_j)	0,1	6.6
proana_sfirst_same(M_i,M_j)	0,1	6.6
proana_pfirst_prev(M_i,M_j)	0,1	6.6
proana_pfirst_same(M_i,M_j)	0,1	6.6
proana_ssubject_prev(M_i,M_j)	0,1	6.6
proana_ssubject_same(M_i,M_j)	0,1	6.6
proana_cb_prev(M_i,M_j)	0,1	6.6
proana_cb_same(M_i,M_j)	0,1	6.6
Additional features for ananphoricity classifier		
same_head_exists(M_i)	0,1	7.2
same_head_distance(M_i)	continuous	7.2
Additional features for antecedenthood classifier		
part_of_apposition(M_j)	nominal	7.2
predicative_NP(M_j)	0,1	7.2
in_negated_clause(M_j)	0,1	7.2
obj_in_modal_clause(M_j)	0,1	7.2

Bibliography

Agirre, E. and G. Rigau (1996). Word sense disambiguation using conceptual density. In *Proceedings of the 16th International Conference on Computational Linguistics*, 16–22.

Ajmera, J., I. McCowan, and H. Bourlard (2004). Robust speaker change detection. *IEEE Signal Processing Letters 11*(8).

Alshawi, H. (ed.) (1992). *The Core Language Engine*. The MIT Press.

Azzam, S., K. Humphreys, and R. Gaizauskas (1998). Coreference resolution in a multilingual information extraction system. In *Proceedings of the Linguistic Coreference Workshop at the International Conference on Language Resources and Evaluation (LREC-1998)*.

Bagga, A. and B. Baldwin (1998a). Algorithms for scoring coreference chains. In *Proceedings of the Linguistic Coreference Workshop at the International Conference on Language Resources and Evaluation (LREC-1998)*, 563–566.

Bagga, A. and B. Baldwin (1998b). How much processing is required for cross-document coreference? In *Proceedings of the Linguistic Coreference Workshop at LREC'98*, 563–566.

Baldwin, B. (1996). Cogniac: A high precision pronoun resolution engine. Technical report, University of Pennsylvania.

Baldwin, B., T. Morton, A. Bagga, J. Baldridge, R. Chandraseker, A. Dimitriadis, K. Snyder, and M. Wolska (1997). Description of the upenn camp system as used for coreference. In *Message Understanding Conference Proceedings*.

Barker, C. and G. K. Pullum (1990). A theory of command relations. *Linguistics and Philosophy 13*, 1–34.

Barzilay, R. and M. Lapata (2005). Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computationa Linguistics*, 141–148.

Bean, D. L. and E. Riloff (1999). Corpus-based identification of non-anaphoric noun phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computationa Linguistics*, 373–380.

Bilenko, M. and R. Mooney (2002). Learning to combine trained distance metrics for duplicate detection in databases. Technical report, Artificial Intelligence Lab, University of Texas at Austin.

Blaheta, D. and E. Charniak (2000). Assigning function tags to parsed text. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 234–240.

Bontcheva, K., M. Dimitrov, D. Maynard, V. Tablan, and H. Cunningham (2002). Shallow methods for named entity coreference resolution. In *Proceedings of the TALN'02: Traitement Automatique des Langues Naturelles*.

Borgman, C. L. and S. L. Siegfried (1992). Getty's synoname and its cousins: A survey of applications of personal name-matching. *Journal of the American Society for Information Science 43*(7), 459–476.

Branting, L. K. (2002). Name-matching algorithms for legal case-management systems. *Journal of Information, Law & Technology* (1).

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

Budanitsky, A. and G. Hirst (2001). Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources*.

Bunescu, R. (2003). Associative anaphora resolution: A web-based approach. In *Proceedings of the EACL 2003 Workshop on the Computational Treatment of Anaphora*.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2*(2), 121–167.

Byron, D. and W. Gegg-Harrison (2004). Eliminating non-referring noun phrases from coreference resolution. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium*.

Carbonell, J. G. and R. D. Brown (1988). Anaphora resoluion: a multi-strategy approach. In *Proceedings of the 12th International Conference on Computational Linguistics*, 96–101.

Cardie, C. and K. Wagstaff (1999). Noun phrase coreference as clustering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 82–89.

Carter, D. (1987). *Interpreting Anaphors in Natural Language Texts*. Ellis Horwood Series in Artificial Intelligence.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 132–139.

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, 115–123.

Cohen, W. W., P. Ravikumar, and S. E. Fienberg (2003). A comparison of string distance metrics for name-matching tasks. In *Proceedings of the KDD Workshop on Data Cleaning and Object Consolidation*.

Collins, M. (1999). *Head-Driven Statistical MOdels for Natural Language Parsing*. Ph. D. thesis, University of Pennsylvania.

Cristea, D., N. Ide, and L. Romary (1998). Veins theory: A model of global discourse cohesion and coherence. In *Proceedings of the 17th International Conference on Computatio nal Linguistics*, 281–285.

Cristea, D., O. Postolache, and R. Mitkov (2002). Handling complex anaphora resolution cases. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium*.

Curran, J. and S. Clark (2003a). Investigating gis and smoothing for maximum entropy taggers. In *Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics*, 91–98.

Curran, J. R. and S. Clark (2003b). Language independent NER using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning*, 164–167.

Dagan, I. and A. Itai (1990). Automatic processing of large corpora for the resolution of anaphora references. In *Proceedings of the 13th International Conference on Computational Linguistics*, 1–3.

Dowty, D., R. Wall, and S. Peters (1981). *Introduction to Montague Semantics*. Kluwer.

Eckert, M. and M. Strube (2001). Dialogue acts, synchronising units and anaphora resolution. *Journal of Semantics 17*(1), 51–89.

Fleischman, M. and E. Hovy (2002). Fine grained classification of named entities. In *Proceedings of the 19th International Conference on Computational Linguistics*.

Fleischman, M. B. and E. Hovy (2004). Multi-document person name resolution. In *Proceedings of the Refrence Resolution Workshop at ACL'04*.

Fox, B. (1987). *Discourse Structure and Anaphora*. Cambridge University Press.

Fraurud, K. (1990). Definiteness and the processing of nps in natural discourse. *Journal of Semantics 7*, 395–433.

Gardent, C. and K. Konrad (1999). Definites or the proper treatment of rabbits. In *Proceedings of the Inference in Computational Semantics Workshop*.

Garrod, S., D. Freudenthal, and E. Boyle (1994). The role of different types of anaphor in the on-line resolution of sentences in a discourse. *Journal of Memory and Language 33*, 39–68.

Gasperin, C., S. Salmon-Alt, and R. Vieira (2004). How useful are similarity word lists for indirect anaphora resolution? In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium*.

Gasperin, C. and R. Vieira (2004). Using word similarity lists for resolving indirect anaphora. In *Proceedings of the Refrence Resolution Workshop at ACL'04*.

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

Givon, T. (ed.) (1983). *Topic Continuity in Discourse: a Quantitative Cross-Language Study*. J. Benjamins.

Gordon, P., B. Grosz, and L. Gilliom (1993). Pronouns, names, and the centering of attention in discourse. *Cognitive Science 17*, 311–348.

Grosz, B. J., A. K. Joshi, and S. Weinstein (1983). Providing a unified account of definite noun phrases in discourse. In *Proceedings of the ?? Annual Meeting of the Association for Computationa Linguistics*, 44–50.

Grosz, B. J., A. K. Joshi, and S. Weinstein (1995). Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics 21*(2), 203–226.

Gundel, J. K., N. Hedberg, and R. Zacharski (1993). Cognitive status and the form of referring expressions in discourse. *Language 69*, 274–307.

Halliday, M. and R. Hasan (1976). *Cohesion in English*. Longman.

Harabagiu, S., R. Bunescu, and S. Maiorano (2001). Text and knowledge mining for coreference resolution. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, 55–62.

Harabagiu, S. and S. Maiorano (1999). Knowledge-lean coreference resolution and its relation to textual cohesion and coherence. In *Proceedings of the ACL Workshop on the Relation of Discourse/Dialogue Structure and Reference*.

Harabagiu, S. and S. Maiorano (2000). Multilingual coreference resolution. In *Proceedings of the Language Technology Joint Conference on Applied Natural Language Processing and the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL2000)*, 142–149.

Hartrumpf, S. (2001). Coreference resolution with syntactico-semantic rules and corpus statistics. In *Proceedings of the 5th Computational Natural Language Learning Workshop (CONLL-2001)*, 137–144.

Hawkins, J. A. (1978). *Definiteness and Indefiniteness: a Study in Reference and Grammaticality Prediction*. London: Croom Helm.

Hayes, P. J. (1981). Anaphora for limited domain systems. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 416–422.

Henschel, R., H. Cheng, and M. Poesio (2000). Pronominalization revisited. In *Proceedings of the 18th International Conference on Computational Linguistics*.

Hirschman, L. and N. Chinchor (1997). MUC-7 coreference task definition. In *Message Understanding Conference Proceedings*.

Hirschman, L., P. Robinson, J. Burger, and M. Vilain (1997). Automating coreference: The role of annotated training data. In *Proceedings of AAAI Symposium on Applying Machine Learning to Discourse Processing*.

Hirst, G. and D. St-Onge (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum (ed.), *WordNet: An Electronic Lexical Database*, 305–332. The MIT Press.

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

Hoste, V. and W. Daelemans (2004). Learning dutch coreference resolution. In *Proceedings of the 15th Computational Linguistics in Netherlands Meeting (CLIN-2004)*.

Hudson, R. (1990). *English Word Grammar*. Basil Blackwell.

Hudson, S., M. Tanenhaus, and G. Dell (1986). The effect of the discourse center on the local coherence of a discourse. In *Proceedings of the 8th Annual Meeting of the Cognitive Science Society*, 96–101.

Jiang, J. J. and D. W. Conrath (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*.

Joachims, T. (1999). Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola (eds), *Advances in Kernel Methods - Support Vector Learning*. MIT-Press.

Joshi, A. and S. Weinstein (1981). Control of inference: Role of some aspects of discourse structure-centering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 435–439.

Kameyama, M. (1985). *Zero Anaphora: The Case of Japanese*. Ph. D. thesis, Stanford University.

Kameyama, M. (1997). Recognizing coreferential links: an information extraction perspective. In *Proceedings of the ACL/EACL Workshop on Operational Factors in Practical, Robust Anaphora Resolution*, 46–53.

Kamp, H. and U. Reyle (1993). *From Discourse to Logic*. D. REidel, Dordrecht.

Karamanis, N. (2003). *Entity Coherence for Descriptive Text Structuring*. Ph. D. thesis, University of Edinburgh.

Karttunen, L. (1976). Discourse referents. In J. McKawley (ed.), *Sytax and Semantics*, Volume 7, 361–385. Academic Press.

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

Kruijff, G.-J. (2002). Formal and computational aspects of dependency grammar.

Kruijff-Korbayova, I. and M. Steedman (2003). Discourse and information structure. *Journal of Logic, Language, and Information 12*, 249–259.

Langacker, R. W. (1969). On pronominalization and the chain of command. In D. Reibel and S. Schane (eds), *Modern Studies in English*, 160–186.

Lapata, M. (2003). Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 545–552.

Lapin, S. and H. J. Leass (1994). An algorithm for pronominal anaphora resolution. *Computational LInguistics 20*(4), 535–561.

Lasnik, H. (1976). Remarks on coreference. *Linguistic Analysis 2*(1).

Latecki, L. (1991). An indexing technique for implementing command relations. In *Proceedings of the 5th Meeting of the European Chapter of the Association for Computational Linguistics*.

Latecki, L. and M. Pinkal (1990). Syntactic and semantic conditions for quantifier scope. In *Proceedings of the Workshop on Plurals and Quantifiers*.

Le, Z. (2004). *Maximum Entropy Modelling Toolkit for Python and C++*.

Leacock, C. and M. Chodorow (1998). Combining local context and wordnet similarity for word sense identification. In C. Fellbaum (ed.), *WordNet: An Electronic Lexical Database*, 265–283. The MIT Press.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on System Documentation*, 24–26.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*.

Llytjós, A. F. (2002). Improving pronunciation accuracy of proper names with language origin classes. In *Proceedings of the European Summer School in Logic, Languge, and Information*.

Loebner, S. (1985). Definites. *Journal of Semantics 4*, 279–326.

Luo, X., A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos (2004). A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computationa Linguistics*.

Mann, G. S. and D. Yarowsky (2003). Unsupervised personal name disambiguation. In *Proceedings of the Conference on Computational Natural Language Learning*.

Mann, W. and S. Thompson (1988). Rhetorical structure theory: Towards a functional theory of text organization. *Text 8*(3), 243–281.

Marcu, D., M. Romera, and E. Amorrortu (1999). Experiments in constructing a corpus of discourse trees: Problems, annotation choices, issues. In *Proceedings of the Workshop on Levels of Representation in Discourse*, 71–78.

Markert, K. and M. Nissim (2003). Corpus-based metonynmy analysis. *Metaphor and Symbol 18*(3), 175–188.

McCallum, A. and B. Wellner (2003). Toward conditional models of identity uncertainty with application to proper noun coreference. In *Proceedings of the IJCAI Workshop on Information Integration on the Web*.

McCarthy, J. and W. Lehnert (1995). Using decision trees for coreference resolution. In *Proceedings of the 14th International Conference on Artificial Intelligence*, 1050–1055.

McKeown, K. (1985). Discourse strategies for generating natural-language text. *Artificial Intelligence 27*(1), 1–41.

Miller, G. (1990). Wordnet: An on-line lexical database. *International Journal of Lexicography*.

Minnen, G., F. Bond, and A. Copestake (2000). Memory-based learning for article generation. In *Proceedings of the Conference on Computational Natural Language Learning*, 43–48.

Mitkov, R. (1997). Factors in anaphora resolution: they are not the only things that matter. a case study based on two different approaches. In *Proceedings of the ACL'97/EACL'97 workshop on Operational factors in practical, robust anaphora resolution*, 14–21.

Mitkov, R. (1998). Robust pronoun resolution with limited knowledge. In *Proceedings of the 18th International Conference on Computational Linguistics*.

Mitkov, R. (1999). Anaphora resolution: the state of the art. Technical report, University of Wolverhampton.

Mitkov, R., R. Evans, and C. Orasan (2002). A new, fully automatic version of mitkov's knowledge-poor pronoun resolution method. In A. Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing*, 169–187. Springer.

Modjeska, N., K. Markert, and M. Nissim (2003). Using the web in machine learning for other-anaphora resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Moser, M. and J. Moore (1996). Toward a synthesis of two accounts of discourse structure. *Computational Linguistics 22*(3), 409–419.

Nenkova, A. and K. McKeown (2003). References to named entities: a corpus study. In *Proceedings of the 4th Meeting of the North American Chapter of the Association for Computational Linguistics*.

Ng, V. (2004). Learning noun phrase anaphoricity to improve coreference resolution: Issues in representation and optimization. In *Proceedings of the 42nd Annual Meeting of the Association for Computationa Linguistics*.

Ng, V. and C. Cardie (2002a). Combining sample selection and error-driven pruning for machine learning of coreference rules. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Ng, V. and C. Cardie (2002b). Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of the 19th International Conference on Computational Linguistics*.

Ng, V. and C. Cardie (2002c). Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computationa Linguistics*, 104–111.

Nissim, M. (2002). *Bridging Definites and Possessives: Distribution of Determiners in Anaphoric Noun Phrases*. Ph. D. thesis, University of Pavia.

NIST (2003). The ace evaluation plan.

Ourioupina, O. (2002). Extracting geographical knowledge from the internet. In *Proceedings of the ICDM-AM International Workshop on Active Mining*.

Pang, B., L. Lee, and S. Vaithyanathan (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Patman, F. and P. Thompson (2003). Names: A new frontier in text mining. In *Proceedings of the 1st NSF/NIJ Symposium*, 27–38.

Pinkal, M. (1991). On the syntactic-semantic analysis of bound anaphora. CLAUS-Report 6.

Poesio, M. (2003). Associative descriptions and salience: a preliminary investigation. In *Proceedings of the EACL Workshop on Anaphora*.

Poesio, M. and M. Alexandrov-Kabadjov (2004). A general-purpose off-the-shelf system for anaphora resolution. In *Proceedings of the Language Resources and Evaluation Conference*.

Poesio, M., S. S. im Walde, and C. Brew (1998). Lexical clustering and definite description interpretation. In *Proceeding of the AAAI Symposium on Learning for Discourse*, 82–89.

Poesio, M., T. Ishikawa, S. S. im Walde, and R. Vieira (2002). Acquiring lexical knowledge for anaphora resolution. In *Proceedings of the Language Resources and Evaluation Conference*.

Poesio, M., R. Stevenson, B. di Eugenio, and J. Hitzeman (2004). Centering: A parametric theory and its instantiations. *Computational Linguistics 30*(3).

Poesio, M., O. Uryupina, R. Vieira, M. Alexandrov-Kabadjov, and R. Goulart (2004). Discourse-new detectors for definite description resolution: a survey and preliminary proposal. In *Proceedings of the Refrence Resolution Workshop at ACL'04*.

Poesio, M. and R. Vieira (1998). A corpus-based investigation of definite description use. *Computational Linguistics 24*(2), 183–216.

Poesio, M., R. Vieira, and S. Teufel (1997). Resolving bridging references in unrestricted text. In *Proceedings of the ACL-97 Workshop on Operational Factors in Practical, Robust, Anaphora Resolution For Unrestricted Texts*, 1–6.

Preiss, J. (2001). Machine learning for anaphora resolution.

Prince, E. E. (1981). Toward a taxonomy of given-new information. In P. Cole (ed.), *Radical Pragmatics*, 223–256. Academic Press.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufman.

Rambow, O. (1993). Pragmatic aspects of scrambling and topicalization in german. In *Proceedings of the Workshop on Centering Theory in Naturally-Ocurring Discourse*.

Reichman, R. (1985). *Getting computers to talk like you and me*. MIT Press.

Reinhart, T. (1983). *Anaphora and Semantic Interpretation*. The University of Chicago Press.

Resnik, P. (1995). Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 448–453.

Reynar, J. C. and A. Ratnaparkhi (1997). A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*.

Sidner, C. L. (1979). *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse*. Ph. D. thesis, MIT.

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

Striegnitz, K. (2004). *Generating Anaphoric Expressions — Contextual Inference in Sentence Planning*. Ph. D. thesis, Saarland University.

Strube, M. (1998). Never look back: An alternative to centering. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics*, 1251–1257.

Strube, M. and U. Hahn (1999). Functional centering — grounding referential coherence in inormation structure. *Computational Linguistics 25*(3), 309–344.

Strube, M., S. Rapp, and C. Müller (2002a). Applying co-training to reference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computationa Linguistics*, 352–359.

Strube, M., S. Rapp, and C. Müller (2002b). The influence of minimum edit distance on reference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 312–319.

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

Tetreault, J. and J. Allen (2004). Dialogue structure and pronoun resolution. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium*.

Uryupina, O. (2003a). High-precision identification of discourse-new and unique noun phrases. In *Proceedings of the ACL'03 Student Workshop*, 80–86.

Uryupina, O. (2003b). Semi-supervised learning of geographical gazetteers from the internet. In *Proceedings of the HLT-NAACL Workshop on the Analysis of Geographic References*.

Uryupina, O. (2004). Linguistically motivated sample selection for coreference resolution. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium*.

Uryupina, O. (2005). Knowledge acquisition for fine-grained named entity classification. In *Proceeding of RANLP*.

van Deemter, K. and R. Kibble (2001). On coreferring: Coreference in muc and related annotation schemes. *Computational Linguistics*.

Vapnik, V. V. (1995). *The Nature of Statistical Learning Theory*. Springer.

Vieira, R. (1998). A review of the linguistic literature on definite descriptions. *Acta Semiotica et Linguistica 7*, 219–258.

Vieira, R. (1999). Applying inductive decision trees in resolution of definite NPs. In *Proceedings of the Argentine Symposium on Artificial Intelligence.*

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

Vilain, M., J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman (1995). A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Message Understanding Conference*, 45–52.

Wagner, R. A. and M. J. Fischer (1974). The string-to-string correction problem. *Journal of the ACM 21*(1), 168–173.

Walker, M., M. Iida, and S. Cote (1994). Japanese discourse and the process of centering. *Computational Linguistics 20*(2).

Wang, G., H. Chen, and H. Atabakhsh (2004). Automatically detecting deceptive criminal identities. *Communications of the Association for Computing Machinery 47*(3), 70–76.

Webber, B. (1979). *A Formal Approach to Discourse Anaphora*. Garland Publishing Inc.

Webber, B., M. Stone, A. Joshi, and A. Knott (2003). Anaphora and discourse structure. *Computational Linguistics 29*(4), 545–587.

Wiebe, J., E. Breck, C. Buckley, C. Cardie, P. Davis, B. Fraser, D. Litman, D. Pierce, E. Riloff, T. Wilson, D. Day, and M. Maybury (2003). Recognizing and organizing opinions expressed in world press. In *Proceedings of the 2003 AAAI Spring Symposium on New Directions in Question Answering*.

Wilks, Y. (1973). Preference semantics. In E. Keenan (ed.), *The Formal Semantics of Natural Language*. Cambridge University Press.

Winkler (1999). The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04.

Yang, X., J. Su, G. Zhou, and C.-L. Tan (2004). Improving pronoun resolution by incorporating coreferential information of candidates. In *Proceedings of the 42nd Annual Meeting of the Association for Computationa Linguistics*.