Rule-based and Memory-based Pronoun Resolution for German: A Comparison and Assessment of Data Sources

von

Holger Wunsch

Philosophische Dissertation
angenommen von der Neuphilologischen Fakultät
der Universität Tübingen
am 19. Mai 2009

Tübingen

2010

Gedruckt mit Genehmigung der Neuphilologischen Fakultät
der Universität Tübingen

Hauptberichterstatter: Prof. Dr. Erhard Hinrichs
Mitberichterstatterin: Prof. Dr. Sandra Kübler
Dekan: Prof. Dr. Joachim Knape

Dank

Mein Dank gilt all jenen, die mich bei der wissenschaftlichen Arbeit und beim Schreiben meiner Dissertation unterstützt haben.

Prof. Erhard Hinrichs danke ich für seine Betreuung. Ohne das wissenschaftliche Umfeld, das er als Leiter des Projekts A1 "Repräsentation und Erschließung linguistischer Daten" des Sonderforschungsbereiches 441 an der Universität Tübingen schuf, wäre es mir nicht möglich gewesen, diese Dissertation zu schreiben.

Ich danke Sandra Kübler, meiner zweiten Betreuerin, und, während ihrer Tübinger Zeit, Kollegin im A1-Projekt. Wenn ich Rat brauchte, konnte ich mich stets auf ihre kompetente und hilfreiche Antwort verlassen.

Aus der Zusammenarbeit mit meinem Projektkollegen Tylman Ule habe ich viel gelernt – seine Art und Weise mit wissenschaftlichen Fragen umzugehen, beeinflussen meine Arbeit bis heute. Das TüPP-D/Z Korpus, zentrale Datenquelle für meine Arbeit, verdanke ich Frank Müller, auch er war ein Kollege im A1-Projekt.

Stephan Kepser hatte stets ein offenes Ohr für mich, und seine wohl überlegten Kommentare brachten wertvolle neue Einsichten. Von Jochen Saile bekam ich manchen guten Rat.

Piklu Gupta und Kathrin Beck lasen große Teile der Dissertation Korrektur, und kommentierten sie sorgfältig – Danke dafür!

Schließlich danke ich meinen Tübinger Freunden, meinen Eltern und meinen beiden Schwestern Charlotte und Friederike – sie sorgten dafür, dass ich auf dieser wissenschaftlichen Fahrt durch Dick und Dünn stets sicher im Wagen sitzen geblieben bin!

meinen Eltern

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The topic of this dissertation is the automatic resolution of third person pronouns in German. As suggested by the term *automatic*, we will consider strategies of practically solving this problem on a computer, instead of developing a normative theory of pronoun resolution. Our main focus of interest will be on issues that directly relate to properties of automatic resolution processes. We will inspect the performance properties of two very different fundamental system designs. The first one is based on a set of linguistic rules, while the second has a machine learning approach at its core. We will further compare the results they produce and discuss their respective properties, advantages, and disadvantages. Central to any approach that is supposed to deal with a linguistic task is the kind of linguistic knowledge that it has access to. Referential relations of pronouns are strongly determined by semantics. As a logical consequence, any automatic approach to pronoun resolution should be provided with comprehensive semantic knowledge. It is a difficult task to come up with a formalized and reliable way of representing semantic information such that it is usable in an automatic approach. We will explore ways of how to incorporate knowledge drawn from lexical semantics and find out whether this knowledge can improve the performance of a resolver. The above constitutes the program of this thesis, as stated in the title:

Rule-based and Memory-based Pronoun Resolution for German:
A Comparison and Assessment of Data Sources

1.1 Cohesion and anaphora

1.1.1 Cohesion

The ultimate goal of any linguistic utterance, regardless whether it is spoken or written, is to convey information of some kind. Of course, the amount of importance that is attributed to the actual information content depends on the type of the message. In a newspaper article, it is clearly the content that is in focus. The structure of the text follows its function, which is to report news. The situation might be different for a poem, for example. Depending on the poet's intentions, the poem's language itself might be in focus. Extreme cases such as this set aside, the intention of a speaker or writer is to pass on information using language as the medium of transportation. In order to maximize the efficiency of this information flow, it is both in the interest of the language producer and the perceiver to minimize the cognitive processing load necessary for understanding the utterance. A very important means to achieve this goal is to arrange the linguistic utterance such that it forms a coherent unity: the more coherent the utterance, the easier it is to understand.

It is cohesion that distinguishes an unstructured bag of words or sentences from a *text* that transports meaning as a whole. In fact, it is the property of being cohesive that Halliday and Hasan (2006) consider the focal and defining property of a text, and they call this property *texture*. Cohesion is a means of establishing texture, or in other words, it is the "semantic glue" that turns a bag of sentences into a text that is meaningful as a whole. Cohesion *semantically relates* two elements in the text to each other. This is achieved by two processes of *presupposition* on the one hand and *satisfaction* on the other hand: If an element is expressed such that its interpretation depends on the meaning of a second element, then it *presupposes* the other element. This other element in turn *satisfies* this presupposition, and this way, a link of meaning is set up between the two. If the presupposed element is missing, it does not satisfy this presupposition. In this case, it is not possible to interpret the second element, and the link between the two elements is broken as well. The semantic links between the elements of a text (Halliday and Hasan (2006) call one such link a *tie*) render the whole utterance coherent and guide the processing of the perceiver by indicating the semantic togetherness of the elements in the text. Halliday and Hasan

(2006) distinguish two different types of cohesion: *Grammatical cohesion*, which is established by syntactico-structural means, and *lexical cohesion*, which is expressed using the lexical properties of the words chosen in the text. The following three sentences,[1] which are a consecutive piece of a contiguous text, illustrate several ties involving different types of cohesion. The elements that set up a tie are marked with square brackets and numbered.

(1) Im Januar hat die [$_1$ **Arbeiterwohlfahrt Bremen]**
In January has the Worker's Welfare Association Bremen
[$_1$ **ihren]** langjährigen Geschäftsführer [$_2$ **Hans Taake]**
its longtime executive Hans Taake
fristlos entlassen, nun wird auch der Vorstand [$_1$ **der**
without notice laid off, now is also the management of the
Wohlfahrtsorganisation] in den Fall hineingezogen.
charity organization in the case drawn in.

'In January, the Worker's Welfare Association laid off its longtime executive Hans Taake without notice, now the management of the charity organization is drawn into the case as well.'

(2) In einer anonymen Anzeige werden der Bremer
In a anonymous complaint are the Bremen
Staatsanwaltschaft Details über dubiose finanzielle Transaktionen
prosecution details about dubious financial transactions
mitgeteilt.
informed.

'In an anonymous complaint, the Bremen prosecution has been informed about dubious financial transactions.'

(3) Verantwortlich, so das Schreiben einer Mitarbeiterin [$_1$ **der AWO]**,
Responsible, so the letter of a employee of the AWO,
sei [$_3$ **die Landesvorsitzende Ute Wedemeier]**, [$_3$ **die]** [$_3$ **sich]** jetzt
is the chairwoman Ute Wedemeier, who herself now
als "Sauberfrau" gebe, "wo doch alle wissen, wie eng
as "Mrs. Clean" gives, "even though everybody knows, how close
[$_3$ **sie]** mit [$_2$ **Taake]** zusammenhing".
she with Taake stuck together".

'The letter of an AWO-employee states that the chairwoman Ute Wedemeier is responsible who now pretends to be "Mrs. Clean, even though everybody knows how close she and Taake stuck together."

[1] The example is taken from the TüBa-D/Z treebank of German, see chapter 5.

The marked elements in the above text correspond to the following ties, which are annotated with their respective types:

Concept 1: *Arbeiterwohlfahrt*

- die Arbeiterwohlfahrt Bremen – ihren *(grammatical)*
- ihren – der Wohlfahrtsorganisation *(grammatical)*
- der Wohlfahrtsorganisation – der AWO *(lexical)*

Concept 2: *Hans Taake*

- Hans Taake – Taake *(lexical)*

Concept 3: *Landesvorsitzende Ute Wedemeier*

- die Landesvorsitzende Ute Wedemeier – die *(grammatical)*
- die – sich *(grammatical)*
- sich – sie *(grammatical)*

We note several points about ties. A tie always holds between two elements in the text. However, ties may form chains of multiple elements, as can be seen in concept 1, which is a chain of the four elements *die Arbeiterwohlfahrt Bremen – ihren – der Wohlfahrtsorganisation – der AWO*. The effect of a chain is that cohesion between meaning-related elements is kept up over a larger span of text.

Language separates in multiple levels, which have been assigned different names in the linguistic literature. But actually, all terms refer to the same concept. Levelt (1991) calls the levels *tiers*, while Halliday and Hasan (2006) use the term *stratum*. They assume three strata, a stratum of sounding or writing, a stratum of wording and finally a stratum of meaning. The strata are ordered hierarchically from lower-level functions of language (phonemes or graphemes) to high-level phenomena such as meaning. Halliday and Hasan emphasize that although cohesion is clearly a semantic relationship, it can be *expressed* with linguistic means anchored in *all* strata. If we go back to the above example, we find that the ties in the first concept *Arbeiterwohlfahrt* are either set up lexically, as in the tie

die Wohlfahrtsorganisation – der AWO, or grammatically, with a pronoun referring back to its antecedent in the pair *die Arbeiterwohlfahrt Bremen – ihren*. Additional examples of grammatical cohesion are structures of coordination, as Halliday and Hasan mention in the following example, where the adjectives *first* and *next* express a relation of temporal sequence:

(4) **First**, he took a piece of string and tied it carefully round the neck of the bottle. **Next**, he passed the other end over a branch and weighted it down with stone.[2]

Irrespective of the precise way the relation is established, grammatically or lexically, the result is a semantic relation of cohesion.

To sum up, cohesion is a central property of text - the "semantic glue" that renders a sequence of sentences into a meaningful whole. A relation of cohesion holds between two elements in a text, but several relations may group into chains, thus keeping up cohesion over longer spans of text. Finally, cohesion can be expressed on different strata of language, but the result is always a semantic relation.

1.1.2 Anaphora and coreference

Anaphora and coreference are cohesive relations, as their purpose is to establish a link of meaning between two elements in a text. Although there are types of anaphora and coreference that do hold between elements other than noun phrases, we will only talk about noun phrases here as this dissertation will be also concerned only with nominal elements. By their nature, anaphora and coreference are special cases of the general concept of cohesion that we discussed above.

In the example from the previous section (1.1.1), we find in terms of anaphora and coreference:

Concept 1: *Arbeiterwohlfahrt*

- die Arbeiterwohlfahrt Bremen – ihren *(anaphoric, grammatical)*
- ihren – der Wohlfahrtsorganisation *(cataphoric, grammatical)*
- der Wohlfahrtsorganisation – der AWO *(coreferential, lexical)*

[2] Halliday and Hasan (2006), p. 13

Concept 2: *Hans Taake*

- Hans Taake – Taake *(coreferential, lexical)*

Concept 3: *Landesvorsitzende Ute Wedemeier*

- die Landesvorsitzende Ute Wedemeier – die *(anaphoric, grammatical)*
- die – sich *(anaphoric, grammatical)*
- sich – sie *(anaphoric, grammatical)*

The example illustrates the characteristic distribution of anaphora and coreference with respect to grammatical and lexical cohesion. Anaphora is a cohesive relation of the grammatical type, while coreference relations are lexical. This can be well explained with specific definitions of anaphora and coreference, such as the one given in van Deemter and Kibble (2001):

Coreference Two elements corefer if they refer to the same extralinguistic referent:

α_1 and α_2 *corefer* if and only if Referent(α_1) = Referent(α_2).

Two elements are coreferent if they refer to the same extralinguistic referent. However, in principle, depending on the way they are expressed, both elements can still be sensibly interpreted even if each one occurs on its own. This is the case if the individual elements are expressed in a way that allows the direct interpretation of their meaning - which is by overt lexical realization. Thus, the tie between two such elements emerges as a consequence of the related meaning of the elements. This is exactly the nature of lexical relations. Thus, if both elements are realized by overt lexical items, coreference relations are cohesive relations of the lexical type.

Van Deemter and Kibble (2001) define anaphora roughly as follows:

Anaphora Two elements α_1 and α_2 are *anaphoric* if α_2 *depends* on α_1 for its interpretation (and cannot be interpreted in the absence of α_1).

Thus for two elements that are in a relation of anaphora, it would be impossible to interpret one of the elements if the other was missing. This element is semantically empty, as is the case for pronouns. A pronoun requires the presence of an antecedent in order to be interpretable. With respect to the way it is expressed, the cohesive relation set up by anaphora is

located on a different stratum than coreference. It is a grammatical relation of cohesion that holds between a purely structural element – the pronoun – and an antecedent which can either be an overtly realized noun phrase or again a pronoun. In the latter case, the antecedent would require another antecedent by itself in order to be interpretable, thus setting up a cohesive (or referential) chain.

It is important to note that the concepts of lexical and grammatical cohesion on the one hand and coreference and anaphora on the other hand are not entirely congruent. While lexical and grammatical relations are disjoint types of cohesion, the terms coreference and anaphora do not denote clearly separable linguistic phenomena: Coreference is *identity of reference*, anaphora means *dependence on an interpretable antecedent*. In fact, most relations that are anaphoric are coreferential as well, since the pronoun and the antecedent refer to the same extralinguistic entity:

(5) Aber [$_1$ **Ercettin]** will Pop mit Niveau machen, sagt [$_1$ **sie]**.
But Ercettin wants pop with quality make, says she.

'But Ercettin wants to make pop with quality, she says.'

The personal pronoun *sie* is anaphoric to the antecedent *Ercettin*, and is only interpretable in presence of the antecedent. But once this interpretation has taken place, it is clear that the pronoun and the antecedent both refer to the same entity, which is the person *Candan Ercettin*. Thus, in addition to being anaphoric, the pronoun and the antecedent are also coreferential.

However, there are cases of anaphora which *do not* involve coreference, as in the following example:

(6) Denn dann hätte [$_1$ **niemand]** gemerkt, daß [$_1$ **er]** in Wahrheit
Because then had nobody noticed, that he in truth
schon verloren war.
already lost was.

'Because then, nobody would have noticed that in truth he was lost already.'

Here, the personal pronoun *er* is anaphoric to *niemand*. However, *niemand* is non-referential, i.e. there is no extralinguistic entity that it refers to. Thus this is an example for a relation that is anaphoric, but not coreferential.

1.2 Anaphora Resolution

The term of *anaphora resolution* is used in different ways. The most general denotation refers to the task of determining the endpoints of a relation of cohesion, and includes coreferential relations, anaphoric relations (in the sense defined in the previous section), and other phenomena, such as the resolution of event anaphora. But the term is also used in a much narrower sense, which is the one that we will adopt in this thesis. In this sense, anaphora resolution is taken to be synonymous to *pronoun resolution*, due to the fact that in many theories of syntax, pronouns as elements that refer back to some other entity, are frequently named *anaphors*. Thus, this thesis is concerned with a task that we define as follows:

Anaphora resolution is the task of finding the correct antecedent for a pronoun.

We will approach the task from the perspective of computational linguistics. Therefore, our main point will be to describe the relevant linguistic issues of the task such that it can be solved *on a computer*. This involves a number of questions:

1. What is the data to work on?
2. What are the typical properties of the data, and what are the consequences of the task?
3. How can the task be formulated as an algorithm?
4. What information does the algorithm require?
5. How well does the algorithm perform?

These questions essentially outline the program that we are going to address in this thesis.

The most important resource in computational linguistics is data. It provides both source of empirical evidence as well as a gold standard against which to verify the results of the linguistic models implemented on the computer. Our source of data in this dissertation will be the *Tübinger Baumbank des Deutschen/Zeitungssprache*, the Tübingen Treebank of Written German (Telljohann et al., 2006), abbreviated *TüBa-D/Z*. It is a collection of articles from the German daily newspaper *die tageszeitung (taz)*, and manually

annotated for syntax, parts of speech and morphology, and referential relations.

Although the *taz* maintains a very colloquial and sometimes inventive style of writing, it still is newspaper text, a genre of mostly reporting, impersonal character. It is characteristic of newspaper text that it rarely contains dialogs and utterances in first or second person. This has a direct consequence on our task, as there will be virtually no pronouns in first or second person. Therefore, we will restrict ourselves to third person pronouns only.

Apart from the data, the algorithm that is chosen is of central importance. We will consider two fundamentally different types of algorithms in this dissertation: The first one is rule-based and could be called "an algorithm proper". Every step that the algorithm takes has been manually defined – so the algorithm very much resembles a set of linguistic constraints or grammar rules put into a procedural format. The advantage of rule-based algorithms is that everything the algorithm does can be inspected and verified, which not only helps to debug the algorithm, but may also give new insights into the problem itself. On the downside, it is a laborious task to implement a rule-based algorithm, and furthermore, if some aspect of the problem is overlooked or unknown, and therefore not stated explicitly as a rule, the algorithm will inevitably fail to handle it.

Machine learning algorithms are designed like a black box: They are presented relevant pre-annotated data, and it is up to the system to induce its own model. Of course, annotating a corpus is an enormous task as well, but unlike a single manually implemented algorithm, the corpus can be reused for very different tasks. However, machine learning approaches may be less accurate as they may induce inaccurate models. We will compare the performance of both a rule-based and a machine-learning-based algorithm on the pronoun resolution task, and we will assess what linguistic information is beneficial for the performance of the algorithms.

Finally, no algorithm will be able to perform properly without the relevant information. As mentioned before, anaphora is mainly a semantic phenomenon, but it is striking that only very little information about semantics has been used in approaches to pronoun resolution in the field of computational linguistics so far. The reason for this is quite simple: Syntactic and morphological data are available on an adequately large scale,

or can be produced even automatically with quite satisfying performance. Semantically annotated corpora however are virtually not available, or not available at sufficiently large scale, which means that this type of data is not (yet) accessible for methods of computational linguistics. We will explore ways to acquire a notion of semantics by automatic data-driven methods and examine whether this additional semantic information can improve the computer's performance.

1.3 Overview of this dissertation

After having sketched the path of research that this dissertation is going to pursue, we will close this introductory chapter with an overview of the chapters that follow.

Chapter 2: Anaphora in Linguistic Theories. In this chapter, we will discuss linguistic theories of anaphora that can be said to be located on opposite sides of the spectrum. *Binding theory* is concerned with sentence anaphora – it describes the restrictions on the distributions of nominals that enter referential relations within the same sentence. *Centering Theory* is a theory that deals with the properties of anaphora when the elements are further apart. This is called *discourse anaphora*.

Chapter 3: Resolution Strategies. This chapter will consider different models of algorithms of pronoun resolution and data structures for the representation of referential relations on a very high and abstract level. We will discuss the first comprehensive rule-based algorithm to pronoun resolution, *Hobbs' algorithm*, and will then contrast the concepts of rule-based algorithms with those of data-driven and machine learning approaches.

Chapter 4: Evaluation Strategies. Without proper evaluation, any research that is based on experiments that generate quantitative results is meaningless. For evaluating the performance of algorithms for anaphora resolution, several strategies have been proposed. We will discuss these and their advantages and disadvantages.

Chapter 5: The Data. In this chapter, we will introduce our two data sources: The manually annotated Tübingen Treebank of Written Ger-

man (TüBa-D/Z), serves as our main data source. The Tübingen Partially Parsed corpus (TüPP-D/Z, Müller 2004b) is a very large, automatically annotated corpus based on the same newspaper source as the TüBa-D/Z corpus. Our data-driven approach for gathering lexical-semantic information uses TüPP-D/Z data.

Chapter 6: Rule-based Approaches. We will dedicate the first part of this chapter to the discussion of two classic rule-based approaches to pronoun resolution which differ in the amount of linguistic information they require. The Resolution of Anaphora Procedure (RAP) by Shalom Lappin and Herbert Leass is a *knowledge-rich* approach. As opposed to this, Christopher Kennedy and Branimir Boguraev's approach is a *knowledge-poor* approach that adopts the same ideas but aims to make do with less sophisticated linguistic information.

In the second part of the chapter, we will introduce our own rule-based implementation of RAP for German.

Chapter 7: Machine-learning-based Approaches. This chapter discusses several approaches to anaphora resolution based on machine learning. Special focus will be on memory-based learning, the machine learning approach that we employ at the core of our machine-learning-based resolver.

Chapter 8: A Hybrid Approach to Pronoun Resolution. In this chapter we will present our hybrid pronoun resolution system, which is a combination of a memory-based resolution module at the core and a number of rule-based pre- and post-processing modules. We will further examine the effect of the structure of the training data on the performance of the resolver.

Chapter 9: Semantics for Pronoun Resolution. In the final chapter, we will be concerned with the question of how to gather semantic information on a data-driven basis for incorporating this into the resolution process and assess the influence of these new semantic features on the performance of the resolver.

Chapter 2

Anaphora in Linguistic Theories

In the previous chapter, we characterized anaphora as a relation of cohesion which semantically connects utterances in a text. Halliday and Hasan (2006) point out that is is not relevant for the effect of cohesion where the two elements between the relations are located. They may be quite far apart, thus establishing coherence over a rather large span of discourse, or they may occur within the same sentence, setting up cohesion in a local domain. About the latter case, Halliday and Hasan write:

> Since the cohesive relations are not concerned with structure, they may be found just as well within a sentence as between sentences. They attract less notice within a sentence, because of the cohesive strength of grammatical structure; since the sentence hangs together already, the cohesion is not needed in order to make it hang together. (Halliday and Hasan, 2006, p. 7f)

While it does not matter for the *semantic effect* of anaphora (or any cohesive relation) where it occurs, location determines important restrictions on the *distribution* of the elements that may enter an anaphoric relation. For two elements that are located further apart (i.e. in two different sentences), the restrictions are fairly weak and mainly determined by semantic factors or processing issues. This is called *discourse anaphora*. However, if both elements occur in the same sentence, there are strong restrictions, which are formulated by the rules of binding theory. This sentence-local type of

anaphora is called *sentence anaphora*. Although they all explain the same problem, different ways of expressing binding theory have been suggested in the linguistic literature. We will discuss three approaches as examples in this chapter: Chomsky (1993) handles anaphora with a configurational, purely syntactic approach as part of his Government and Binding Theory. Pollard and Sag (1994) present a largely non-configurational analysis of binding theory based on the obliqueness hierarchy of subcategorization within their HPSG framework. A semantic formulation of binding theory based on the hierarchy of thematic roles is given in Jackendoff (1972).

Theories that describe discourse anaphora are for example Discourse Representation Theory (Kamp, 1981; Kamp and Reyle, 1993), which we are not going to discuss any further, or Centering Theory (Grosz et al., 1995). Centering Theory perceives the factors that control anaphora as mainly determined by the need to minimize cognitive load on both the language producer and the language perceiver with the goal of optimizing information conveyance.

2.1 Binding theory

Binding theory is in the simplest case concerned with syntactic configurations of the following kind:

(1) a. John$_i$ hates himself$_i$.
b. *John$_i$ hates himself$_j$.
c. *John$_i$ hates him$_i$.
d. John$_i$ hates him$_j$.

In both (1-a) and (1-b), the object of the verb is the reflexive *himself*. In (1-a), *himself* is co-indexed with the subject *John*, which indicates identity of reference. In (1-b), *himself* is *not* co-indexed with *John*, denoted by the different index j. While (1-a) is grammatical, (1-b) is not.

The situation is the other way round when the reflexive is replaced by the personal pronoun *him*, as in (1-c) and (1-d). Now the sentence becomes ungrammatical when the pronoun is co-indexed with *John* as in (1-c), and is grammatical when the pronoun is not co-indexed, as in (1-d).

The third category that binding theory deals with are full NPs (also called definite descriptions) as in the following example:[1]

(2) a. She$_i$ admires Mary$_j$.
b. *She$_i$ admires Mary$_i$.

In (2-a), the full NP *Mary* is not co-indexed with the subject *she*, resulting in a grammatical sentence. In (2-b), however, co-indexing *Mary* with *she*, meaning that *she* and *Mary* refer to the very same person, renders the sentence ungrammatical. Note that the same is true for non-pronominal subjects such as

(3) a. The lady$_i$ admires Mary$_j$.
b. *The lady$_i$ admires Mary$_i$.

Already these simplistic examples illustrate a core observation of binding theory. Looking at the distribution of reflexives and personal pronouns in example (1) it is obvious that the pronouns are *complementarily* distributed: where it is grammatical to use a reflexive, using a personal pronoun is ungrammatical, and vice versa. This complementary distribution of pronouns has been systematically described as early as by Lees and Klima (1969) (the cited article was originally published in 1963), later for example by Postal (1966), and by Lasnik (1989).

2.1.1 Terminology

In binding theory, nominals are subdivided in three basic relevant categories. In different formulations of binding theory, different terms have been coined, some of which are slightly confusing. Büring (2005) suggests the following clearly distinguished set of categories:

Reflexives and reciprocals: reflexive pronouns such as *himself, herself, oneself*, or the German *sich*, as well as reciprocals such as *each other* or, in German, *einander*

Non-reflexive pronouns comprise the class of personal pronouns. (*she, her, it, he, him* etc., *sie, es, ihn, uns* etc.)

[1] Binding theory also applies to empty categories, such as traces. Since this aspect of binding theory is not relevant for the present problem setting, we will not consider it any further.

Full NPs including names: Names, common nouns, and quantifiers such as *Mary, everyone, tree* (German: *Maria, jede(r), Baum*), and so on. Chomsky (1993) defines them as including "... noun phrases with heads that are in some intuitive sense 'potentially referential'..." (p. 102).

2.2 The treatment of anaphora in Government and Binding

As part of his *Government and Binding Theory*, Chomsky (1993) developed a configurational approach to the treatment of the linguistic phenomenon of anaphora. Chomsky's variant of binding theory devises constraints on the syntactic conditions under which a noun phrase may function as the antecedent of a pronoun.

Since GB's terminology deviates to some extent from what was introduced above and what is used in the specific field of pronoun resolution, it seems appropriate on this occasion to include a few remarks about GB's terminology. *Anaphors*, in GB terms, comprise the reflexives and reciprocals. It is important not to confuse this term with the relation of *anaphora* which denotes a referential relation between a nominal element α and an antecedent β where the reference of α is dependent on and can be only interpreted in presence of β.[2] Specifically, the possibility of entering an anaphoric relation with an antecedent is not restricted to reflexives (or reciprocals) as GB's label 'anaphor' might imply: personal pronouns, which are called *pronominals* in GB, may enter anaphoric relations as well. *R-Expressions* finally conflate with *full NPs* – noun phrases with independent overt reference.

In the following, we will present Chomsky's binding theory in a rather compressed form.

The core concept in GB binding theory is a structural syntactic relation between two nodes in a tree, which is called *binding*. This relation is formulated in terms of the *c-command* relation, which is defined as follows:[3]

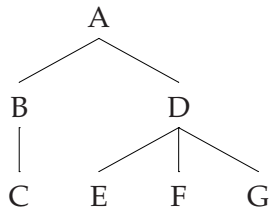
[2] See section 1.1.2 for a discussion of the terms *anaphora* and *coreference*.

[3] We cite the definition as given in von Stechow and Sternefeld (1988), p. 293. The c-command relation was first introduced by Reinhart (1976).

C-Command

α **c-commands** β if and only if there exists a configuration $[_\gamma \ldots \alpha \ldots \beta \ldots]$ or $[_\gamma \ldots \beta \ldots \alpha \ldots]$ where γ is the next branching node that dominates α.

The following example illustrates the c-command relation.



The table shows all c-command relations that hold in the example tree. The c-command relation is not symmetric, therefore the table can only be read in one direction: the nodes in columns c-command the nodes in rows. So E c-commands G, and D c-commands B.[4]

	A	**B**	**C**	**D**	**E**	**F**	**G**
A							
B				X	X	X	X
C				X	X	X	X
D		X	X				
E						X	X
F					X		X
G					X	X	

The binding relation is then defined as follows:[5]

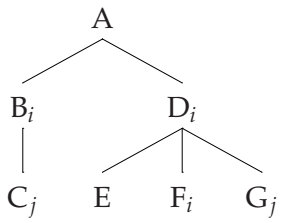
Binding Relation

1. α is **bound** by β if and only if α and β are co-indexed and β c-commands α.
2. α is **free** if and only if it is not bound.

[4] Reinhart (1983) points out that assuming that the domination relation is reflexive, all nodes would also c-command their parent nodes, and furthermore themselves.

[5] We present a simplified version of the full binding theory here and omit the definition of binding in argument and non-argument positions as well as local binding and binding of variables.

Thus, binding is c-command plus co-indexation. In the example tree below, some nodes are co-indexed, yielding several binding relations:



The binding relations are:

- A is free, because it is neither c-commanded nor co-indexed with any node.
- B is bound by D. D c-commands B, and B is co-indexed with D.
- C is free (it is not c-commanded).
- D is bound by B. B c-commands D, and D is co-indexed with B.
- E is free (it is not co-indexed with any node).
- F is bound by B. B c-commands F, and F is co-indexed with B.
- G is bound by C. C c-commands G, and G is co-indexed with C.

Based on the above relations, the definition of binding theory is:

Binding Theory

(A) An anaphor is bound in its binding category

(B) A pronominal is free in its binding category

(C) An R-expression is free.

This definition depends on the concept of a *binding category*, which can be defined as follows.

Binding Category

The **binding category** of α is the smallest category γ such that

(i) γ dominates a SUBJECT β which c-commands α, and

(ii) β is accessible to α if α is an anaphor.

This again depends on two further concepts, the "big SUBJECT", and the accessibility of a SUBJECT.

"Big" SUBJECT

The **SUBJECT** of α is

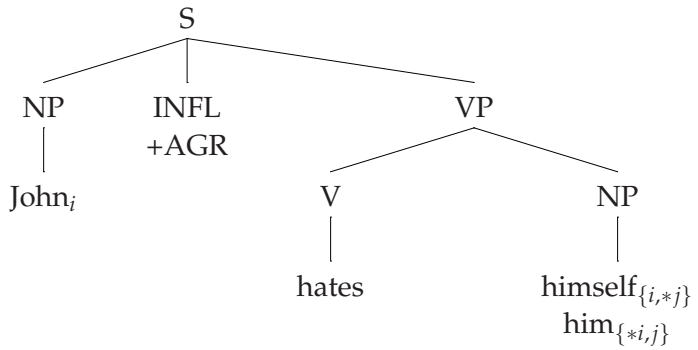
1. AGR, if α is a finite clause
2. the (ordinary) subject of α otherwise

Accessible SUBJECT

1. **i-over-i-Filter:** *$[_\gamma \ldots \delta \ldots]$ where γ and δ bear the same index.
2. A SUBJECT α is **accessible** to β if and only if α c-commands β and assignment to β of the index of α would not violate the i-over-i-Filter.

Coming back to the examples at the beginning of this section, using these structural relations, binding theory predicts the grammaticality or ungrammaticality of the respective syntactic configurations.

The sentences in example (1) have the simple syntactic structure shown below.



The binding category of *himself/him* is the whole sentence. The INFL node in S is the SUBJECT, and, for the anaphor *himself*, it is accessible (for the pronominal *him*, the accessibility requirement does not apply).

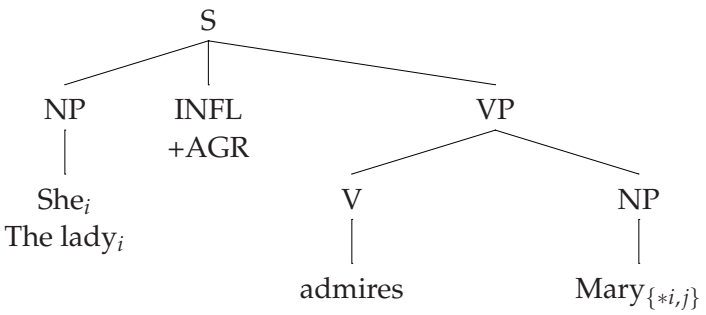
The subject *John*$_i$ binds the pronoun *himself*$_i$ (note the index *i*). The binding category of *himself*$_i$ is the whole sentence, with AGR as the accessible SUBJECT. Principle A requires anaphors to be bound, therefore binding theory licenses this sentence as grammatical.

With index j, *himself*$_j$ is free. This violates Principle A, and in fact, the corresponding reading is ungrammatical.

For the personal pronoun (pronominal) *him*, not surprisingly, the situation is reversed. Indexed i, *him*$_i$ is bound by *John*$_i$ in the same binding category as *himself*. This violates Principle B, which requires pronominals to be free in their binding category. Therefore, binding theory correctly rejects the sentence as ungrammatical.

Finally, *him*$_j$ is not co-indexed with *John*$_i$. Therefore, it is free in its binding category. Binding theory correctly predicts that the sentence is grammatical.

The relevant structures are largely the same for the R-expressions in examples (2) and (3):



Co-indexing *Mary* with *she*$_i$ or *the lady*$_i$, respectively, would result in binding of *Mary*$_i$, which is not allowed by Principle (C). Therefore, for the sentence to be grammatical, *Mary* must be free (not co-indexed).

This section closes with a slightly more complex example to illustrate some more of the effects of the constraints stated in binding theory.

(4) [$_{S_1}$ [The children]$_i$ AGR thought that [$_{S_2}$ [pictures of each other$_i$] AGR were on sale]]]

This sentence is correctly licensed. There are three potential binding domains:

1. the NP *pictures of each other*, with *pictures* as the antecedent. In this case, the co-indexing would be

   [pictures$_i$ of each other$_i$]$_i$

   with *pictures* co-indexed with the whole NP (since it is the head). This

is a violation of the i-over-i filter, which is why this possibility is ruled out.

2. the embedded clause *pictures of each other were on sale*, with co-indexing:

   [pictures of each other$_i$]$_i$ AGR$_i$ were on sale

   and AGR as accessible SUBJECT. However, again, this would be a violation of the i-over-i-filter.

3. finally, the complete sentence *the children thought that pictures of each other were on sale*, with co-indexing:

   [The children]$_i$ AGR thought that pictures of [each other]$_i$ were on sale.

   which is admitted by principle A, since *the children* binds *each other*.

However, Pollard and Sag (1994) note that the structurally equivalent correct sentence:

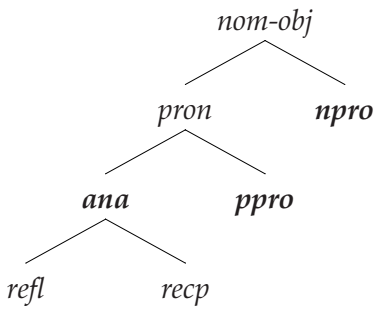
(5) John suggested that [portraits of [each other]$_i$ would amuse [the twins]$_i$]

is incorrectly ruled out by Chomsky's approach, since *the twins* fails to c-command *each other*, thus the reciprocal is free, violating principle A. The HPSG binding theory which we going to discuss in the next section provides a solution to this issue.

2.3 Binding theory within the HPSG framework

Pollard and Sag (1994) present a non-configurational account of binding in their framework of Head-Driven Phrase Structure Grammar. HPSG binding theory does not rely on tree configurations, but is formulated as principles (or constraints) on the obliqueness of elements on the SUBCAT lists of lexical heads.

The three classes of overt nominals that we introduced in section 2.1.1 (reflexives and reciprocals, non-reflexive pronouns and full NPs) have direct counterparts in the HPSG sort hierarchy:



The sort *ana ("anaphors")* with subsorts *refl* and *recp* comprises the reflexives and reciprocals. The sort *ppro* represents the non-reflexive pronouns, and finally *npro* are the full NPs. Thus, HPSG integrates a classification of the three types of overt nominals that are relevant for binding theory in the CONTENT value of any nominal object. The CONTENT value of the reflexive pronoun *herself* is as follows:

$$refl\begin{bmatrix} \text{INDEX} & ref\begin{bmatrix} \text{PER} & 3rd \\ \text{NUM} & sing \\ \text{GEND} & fem \end{bmatrix} \\ \text{RESTR} & \{\} \end{bmatrix}$$

whereas the personal pronoun *they* has the CONTENT value of:

$$ppro\begin{bmatrix} \text{INDEX} & ref\begin{bmatrix} \text{PER} & 3rd \\ \text{NUM} & plur \end{bmatrix} \\ \text{RESTR} & \{\} \end{bmatrix}$$

Note the sort labels *refl* and *ppro* on the respective fragments.

Given this, Pollard and Sag (1994) define a new relation of *local obliqueness command* as follows:

Local O-Command

Let Y and Z be *synsem* objects with distinct LOCAL values, Y referential. Then Y *locally o-commands* Z just in case Y is less oblique than Z.

The local o-command principle operates on the sequence of *synsem* objects that occur on the SUBCAT list of a lexical head.

In the sentence

(6) John hates himself.

the SUBCAT list of the verb *hates* looks as follows:

$$\left[\text{SUBCAT} \quad \left\langle \text{NP:}\textit{npro}, \text{NP:}\textit{ana} \right\rangle\right]$$

John is less oblique than *himself*, and referential, therefore *John* locally o-commands *himself*.

It is obvious that the non-configurational relation of *local o-command* is the equivalent to c-command in configurational analyses. Where binding is defined there as "c-command plus co-indexation", this is defined analogously in HPSG as "o-command plus co-indexation":

Local O-Binding

Y *locally o-binds* Z just in case Y and Z are co-indexed and Y locally o-commands Z. If Z is not locally o-bound, then it is said to be *locally o-free*.

Furthermore, Pollard and Sag (1994) define the more general relation of *o-command* as follows:

O-Command

Let Y and Z be *synsem* objects with distinct LOCAL values, Y referential. Then Y *o-commands* Z just in case Y locally o-commands X dominating Z.

Unlike local o-command, the general o-command is dependent on tree configurations, as it makes use of the domination relation.[6]

In a manner very similar to the configurational variant, the HPSG binding theory is defined as follows:

HPSG Binding Theory

Principle A: A locally o-commanded anaphor must be locally o-bound.
Principle B: A personal pronoun must be locally o-free.
Principle C: A nonpronoun must be o-free.

[6] Pollard and Sag (1994) suggest a fully non-configurational variant of binding theory, the discussion of which would be beyond the scope of this dissertation. The reader is referred to the HPSG book.

The simple examples from the introduction would be treated as follows:

(7) John$_{\boxed{1}}$ hates himself$_{\boxed{1}}$

(8) *John$_{\boxed{1}}$ hates himself$_{\boxed{2}}$

In example (7), *John* is co-indexed[7] with *himself*. Since John locally o-commands *himself*, it also locally o-binds the anaphor, therefore, principle A is satisfied. In example (8), *himself* is locally o-free since it is not co-indexed with *John*, a violation of principle A.

2.3.1 Exempt anaphors

Due to the formulation of the binding principles in HPSG theory, some anaphors are *exempt* from being restricted by the principles, if they occur in a position on a subcat list which is not o-commanded by another element. One example for this is

(9) *Himself$_i$ has eaten

Principle A of HPSG binding theory does not apply here, *himself* is the only element on the subcat list of *eaten* and therefore not o-commanded. Pollard and Sag (1994) argue that this sentence is ruled out for a different reason: The reflexive *himself* only has an accusative form, while in the subject position, nominative is required. Nonfinite clauses, which require an accusative subject are not ruled out:

(10) John$_i$ wanted more than anything else for **himself**$_i$ to get the job.

Pollard and Sag further argue that in

(11) [What John$_i$ would prefer] is for himself$_i$ to get the job.

the indicated co-indexing seems just as obligatory as in (10). This sentence is incorrectly ruled out by Chomsky's binding theory, since *John* is embedded in the subject clause and does not c-command *himself*. In the HPSG account, *himself* is exempt from principle A, therefore, the sentence is licensed.

[7] We use HPSG-style notation of co-indexing in boxes here, indicating that co-indexation is structure sharing of the INDEX value in the CONTENT of a sign.

Example (5) from the previous section, repeated here

(12) John suggested that [portraits of [each other]$_i$ would amuse [the twins]$_i$]

which is ruled out by Chomsky's approach, is correctly licensed, since *each other* is again exempt from principle A. Pollard and Sag (1994) argue that the exempt cases are not to be accounted for by the binding principles, but by other external factors (Pollard and Sag, 1994, p. 266 ff.)

2.4 A semantic formulation of binding theory based on theta-roles

We discussed two variants of formulating binding principles so far. The first one, Chomsky's GB theory, is a purely syntactic, configurational approach which is based on the c-command relation, a relation on the structure of a syntactic tree. The second is the non-configurational binding theory which is part of the HPSG framework (Pollard and Sag, 1994). Its core relation is o-command, which is expressed in terms of an obliqueness hierarchy of grammatical functions of the elements on the SUBCAT lists of lexical heads.

A third way of implementing binding theory is by means of the ranked hierarchy of thematic roles, as described by Jackendoff (1972). The usage of thematic roles makes this version of binding theory a semantic one. Jackendoff assumes the following hierarchy of thematic roles:

1. AGENT
2. LOCATION, SOURCE, GOAL
3. THEME

and then formulates the following thematic hierarchy condition on reflexives, which we may call "Θ-Principle A".

Θ-Principle A

A reflexive may not be higher on the Thematic Hierarchy than its antecedent.

This statement is of course equivalent to the other syntactic formulations of Principle A. Thus, the sentence

(13) John$_i$ hates himself$_i$.

is valid according to the principle A, since *John* has the thematic role of AGENT and *himself* has the role of THEME. Büring (2005) mentions that an advantage of the theta-role based binding theory is that it immediately applies to PP complements such as in

(14) a. We talked to John$_i$ about himself$_i$.
b. *We talked to himself$_i$ about John$_i$.

In (14-a) *John* has the role of the GOAL, and *himself* is the THEME, where GOAL is higher in the hierarchy than THEME. In other words, *John* "Θ-commands" *himself* (Büring, 2005), and Θ-Principle A is satisfied. In (14-b), the roles are swapped, which violates Principle A and renders the sentence unacceptable.

Wilkins (1988) points out that Jackendoff's account can't properly explain the difference between

(15) a. We left the child by herself.
b. *We left herself by the child.

In (15-a), the reflexive *herself* has the role LOCATION, and *child* is the *Theme*. The thematic role of *herself* is thus ranked higher in the hierarchy than its antecedent *child*, which would be a violation of Θ-Principle A, and therefore (incorrectly) ruled out. Wilkins argues that argument *child* in (15-a), which is *directly* assigned a thematic role by the verb instead of *indirectly* by a preposition, has two thematic roles: THEME and PATIENT, and suggests an extended role hierarchy:

1. AGENT
2. PATIENT
3. LOCATION, SOURCE, GOAL
4. THEME

In this way, the thematic role of *child* in (15-a) is higher than *herself*, rendering the sentence grammatical.

It is not our goal to provide a full account of all the "gory details" of binding theory. Each of the accounts has its specific strengths and weaknesses, however, it is beyond the scope of this work to cover the advanced cases and discuss the problematic cases. Here, we refer the reader to the rich primary literature on binding theory.

2.5 Centering Theory

In the previous sections, we briefly introduced three representatives of binding theory. Binding theory describes the phenomenon of *sentence anaphora*, i.e. principles and restrictions under which a relation of anaphora can be established between elements *within the same sentence*.

Centering Theory (Grosz et al., 1995) is concerned with *discourse anaphora*, i.e. it considers the phenomenon from a discourse-wide point of view. It is based on the insight that the task of understanding and following a discourse requires considerable cognitive effort on the part of the hearer. Centering Theory claims that the higher the processing load, the harder it becomes for the hearer to understand a discourse. Conversely, if the processing load is minimized, understanding a discourse is easier. The theory formulates rules and conditions which model the amount of cognitive effort that is required for a hearer to understand the discourse.

In Centering Theory, the relevant linguistic entity is a discourse. Centering Theory fits within the general theory of discourse structure, which was developed by Grosz and Sidner (1986) before Centering Theory was devised by Grosz et al. (1995). Grosz and Sidner (1986) distinguish three components of discourse:

- At the level of **linguistic structure**, a discourse is separated into *discourse segments*. A discourse segment is subdivided into multiple *utterances*, which are *coherent* to a certain degree. Coherence between utterances is called *local coherence*.

  Furthermore, discourse segments exhibit *global coherence*, which is coherence with other segments in the discourse.

- The level of **intentional structure** comprises intentions and relations among them: *"The intentions provide the basic rationale for the discourse,*

and relations represent the connections among these intentions."
(Grosz et al. 1995, p. 204)

- The **attentional state** models the discourse participants' focus of attention at any point in the discourse.

If the concept of binding is the central concept of binding theory, the central concept in Centering Theory is that of *coherence*. The core claim of Centering Theory is that the coherence of a discourse determines the processing load the discourse puts on the participants. The more coherent the discourse, the smaller the processing load. The less coherent the discourse, the higher the processing load. Note that this dependence can be understood in the opposite direction as well: The higher the processing load, the more cognitive effort is required on the part of a participant, and the harder it is to follow the discourse. As the result, the discourse is rendered incoherent.

Grosz et al. (1995) illustrate the effects of coherence in a discourse with the following two short examples:

(16) 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.

(17) a. John went to his favorite music store to buy a piano.
b. It was a store John had frequented for many years.
c. He was excited that he could finally buy a piano.
d. It was closing just as John arrived.

Both discourse (16) and (17) convey exactly the same information. However, discourse (16) is intuitively more coherent than discourse (17). With everything else the same, the reason for the difference must be found in sentences (b) and (d) of both discourses.

Discourse (16) is clearly all about *John* – sentences (a) to (d) remain focused on *John*, with the subject being realized as a pronoun in sentences (b) to (d). This situation is graphically depicted in figure 2.1-a. There are two entities (of relevance to this presentation), *John*, and the *music store*, but *John* remains the prominent entity in all four sentences.

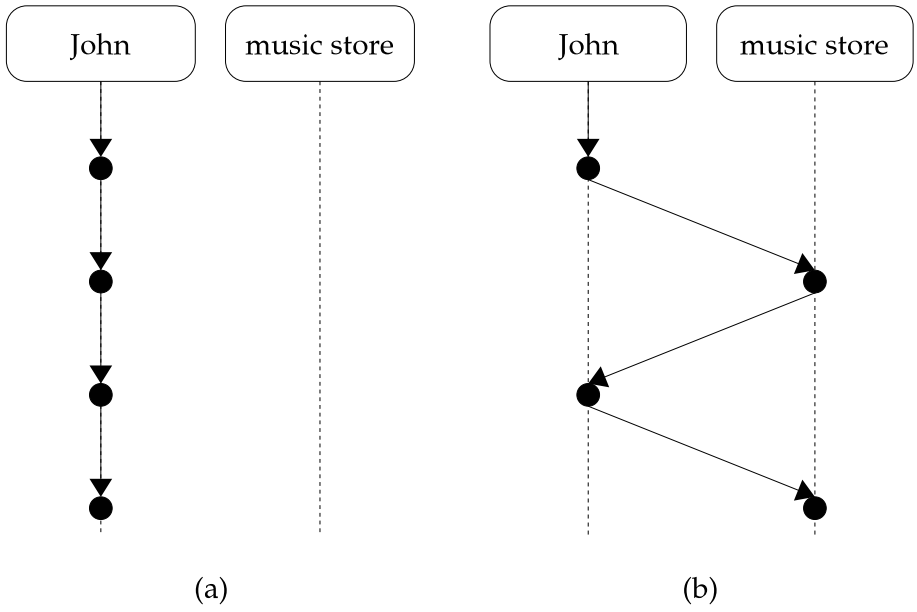


Figure 2.1: Focused entity remains constant (a) – Focused entity flips back and forth (b)

In discourse (17), the situation is different. In (17-a), the prominent entity is *John*. In (17-b), however, the *music store* appears in the subject position in pronominalized form, which indicates to the hearer that this entity is currently in focus. In fact, *John* follows as definite NP. Thus, from (17-a) to (17-b), the focus is shifted from *John* to the *music store*. In (17-c), the focus flips back to *John*, which is realized as pronoun. In (17-d) finally, another shift occurs. Centering Theory states that this flipping back and forth (which is illustrated in figure 2.1-b) increases the inference load placed upon the hearer, resulting in the perceived loss of coherence.

2.5.1 Centers

In Centering Theory, the entities that establish coherence between utterances, the "glue" of discourse, so to speak, are called the *centers* of an utterance, and the process of selecting centers is called *centering*, which is what gave Centering Theory its name.

Grosz et al. (1995) state that centers are a property of utterances, and not sentences – one and the same sentence may have different centers when

uttered in different discourses: *"Centers are thus discourse constructs. Furthermore, centers are semantic objects, not words, phrases, or syntactic forms"* (Grosz et al., p. 208).

Centers are related to linguistic expressions by means of a semantic relation of *realization*. According to Grosz et al., this relation combines syntactic, semantic, discourse, and intentional features.

Centers connect an utterance in a discourse segment to the next. An utterance U_n is assigned a partially ordered set of *forward-looking centers*, labeled $C_f(U_n)$, and exactly one *backward-looking center* $C_b(U_n)$. The set of forward-looking centers contains those entities that are the most prominent in an utterance U. They are ordered according to their likelihood of becoming the single *backward-looking center* $C_b(U_{n+1})$ in the next utterance.[8] The most highly ranked element of $C_f(U_n)$ that is realized in U_{n+1} becomes the $C_b(U_{n+1})$.

The introductory example, which we repeat here, illustrates the interplay of the forward and backward looking centers:

(18) a. John went to his favorite music store to buy a piano.
b. He had frequented the store for many years.

Sentence (18-a) contains three somewhat prominent concepts: *John*, *music store*, and *piano*. They constitute the set of forward-looking centers, i.e. the ordered set of concepts that are likely to occur as the focused concept in the next utterance. Here, we would have:

C_f(18-a) = {*John*, *music store*, *piano*}, where *John* $\succ$ *music store* $\succ$ *piano*.

(18-b) continues the discourse with the pronoun *he* in subject position, referring to *John*. Pronominalization is a very frequent means of focusing a referent, and the fact that the pronoun is additionally in the subject position renders *John* the most prominent concept in (18-b), and the backward-looking center C_b(18-b).

The ordering of the forward-looking centers is a matter of interpretation. The one suggested here is certainly the most likely one, but, depend-

[8] Grosz et al. (1995) do not strictly prescribe according to what criteria the lists of forward-looking centers $C_f(U_n)$ are ordered (but see the end of section 2.5.2). In actual implementations, the obliqueness of argument roles can be used (Beaver, 2004). In their extension to Centering Theory called *Functional Centering*, Strube and Hahn (1999) ranked the C_f list according to the information structure of the utterance.

ing on the context in which the utterance occurs (that is, ultimately depending on the speaker's intentions), other sequences might be conceivable as well.

Centering Theory defines three translation relations that hold between a pair of utterances:

1. **Center Continuation:** $C_b(U_{n+1}) = C_b(U_n)$, and this entity is the most highly ranked element of $C_f(U_{n+1})$. In this case, $C_b(U_{n+1})$ is the most likely candidate for $C_b(U_{n+2})$; it continues to be C_b in U_{n+1}, and continues to be likely to fill that role in U_{n+2}.

2. **Center Retaining:** $C_b(U_{n+1}) = C_b(U_n)$, but this entity is not the most highly ranked element of $C_f(U_{n+1})$. In this case, $C_b(U_{n+1})$ is not the most likely candidate for $C_b(U_{n+2})$; although it is retained as C_b in U_{n+1}, it is not likely to fill that role in U_{n+2}.

3. **Center Shifting:** $C_b(U_{n+1}) \neq C_b(U_n)$.

The translation relations are illustrated in figure 2.2.

	$C_b(U_n) = C_b(U_{n-1})$	$C_b(U_n) \neq C_b(U_{n-1})$
$C_b(U_n) = C_p(U_n)$	**Continuation**	**Shifting**
$C_b(U_n) \neq C_p(U_n)$	**Retaining**	

Figure 2.2: Translation relations. $C_p(U_n)$ is the preferred center (taken from Brennan et al. (1987)).

2.5.2 Centering rules

The coherence of a discourse is determined by the types of translation relations that a speaker selects. The more frequently centers are shifted, the less coherent a discourse becomes. Centering Theory defines two rules that model the influence of the choice of centers on the coherence of a discourse.

Rule 1

If any element of $C_f(U_n)$ is realized by a pronoun in U_{n+1}, then the $C_b(U_{n+1})$ must be realized by a pronoun as well.

This rule states two requirements about centers:

- If any forward-looking center from U_n is realized as a pronoun in U_{n+1}, then the backward-looking center in U_{n+1} must be realized as a pronoun as well.
- No pronoun can occur in U_{n+1} unless the backward-looking center in U_{n+1} is realized as a pronoun.

An example for the way this rule works follows below.

Rule 2

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

This reflects the intuition that in a discourse the shift of a center should be smoothly introduced, i.e. the hearer should be provided hints that a center shift is going to occur soon. A coherent discourse therefore consists of spans of center continuations followed by center retainments (which rerank the C_f), and then finally by a center shift which marks the new center.

The following examples (taken from Grosz et al.) give an intuition on how the centering rules work. For the first example it is assumed that it is part of a larger discourse with *John* currently being centered:

(19) a. He has been acting quite odd. [C_b = *John*, referent(*he*) = *John*]
b. He called up Mike yesterday. [C_b = *John*, referent(*he*) = *John*]
c. John wanted to meet him urgently. [C_b = *John*, referent(*him*) = ***Mike***]

Utterance (19-c) renders the discourse incoherent. While the backward-looking center *John* is realized by a definite NP, *Mike* is realized by a pronoun. This violates rule 1. Grosz et al. (1995) state that the only way to sensibly interpret (19-c) is that a person named John has been newly introduced into the discourse which is different from *John* that is currently centered.

Figure 2.3 illustrates this graphically. The backward-looking centers $C_b(U_k)$ are represented by the solid black dots. The thin circles around the black dots indicate which entity is *linguistically expressed* to be currently in focus. The backward-looking center C_b is *John* in all three utterances (a) to (c). This is indicated by the black dots that stay on the left line. In (a)

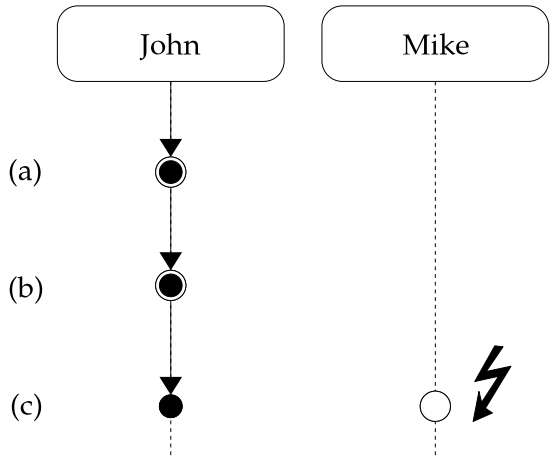


Figure 2.3: Violation of rule 1 (see text for explanation)

and (b), *John* is also linguistically marked as being in focus, by using the pronoun *he*. Therefore, a circle is placed around each dot. However, in (c), due to the pronominalization of *Mike*, the factual backward-looking center *(John)* and its linguistic realization (pronominalization of *Mike*) suddenly diverge. This is depicted by the circle which has moved to the right line, while the black dot stays behind on the left. The coherence of this example discourse suffers because in (c), the speaker centered a different entity without any prior cues of his intention to do so.

By introducing an additional utterance that explicitly *shifts* the center from *John* to *Mike*, the coherence of this discourse can be re-established:

(20) a. John has been acting quite odd.
b. He called up Mike yesterday. [C_b = *John*, referent(*he*) = *John*]
c. Mike was studying for his driver's test. [C_b = *Mike*, referent(*his*) = *Mike*]
d. He was annoyed by John's call. [C_b = *Mike*, referent(*he*) = *Mike*]

Utterance (c) introduces a center shift which moves the current center away from *John* to *Mike*, resulting in a much more coherent discourse. Figure 2.4 illustrates this.

The slightly more complex example below illustrates the interaction of centering rules forming a coherent discourse.

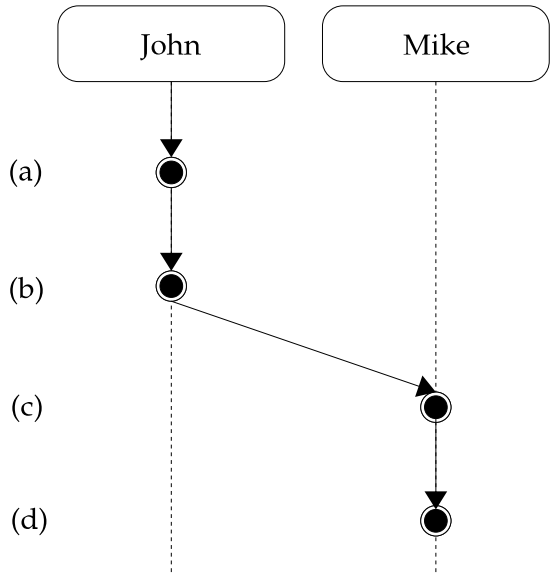


Figure 2.4: Center shifting

(21) a. John has been having a lot of trouble arranging his vacation.
b. He cannot find anyone to take over his responsibilities.
(*he* = *John*)
C_b = *John*; C_f = {*John*}
c. He called up Mike yesterday to work out a plan.
(*he* = *John*)
C_b = *John*; C_f = {*John, Mike*} **continue**
d. Mike has annoyed him a lot recently.
(*he* = *John*)
C_b = *John*; C_f = {*Mike, John*} **retain**
e. He called John at 5 am on Friday last week.
(*he* = *Mike*)
C_b = *Mike*; C_f = {*Mike, John*} **shift**

Figure 2.5 illustrates example (21) graphically. The interesting translation is the retain between (c) and (d). Although the backward-looking center still remains the same *(John)*, it is unlikely that it will do so in following utterances: The focus already moves away from *John* to *Mike*, which is depicted by the dashed line and the gray dot. From utterance (d) to (e), the expected shift occurs. The "announced" type of shift in (21) pro-

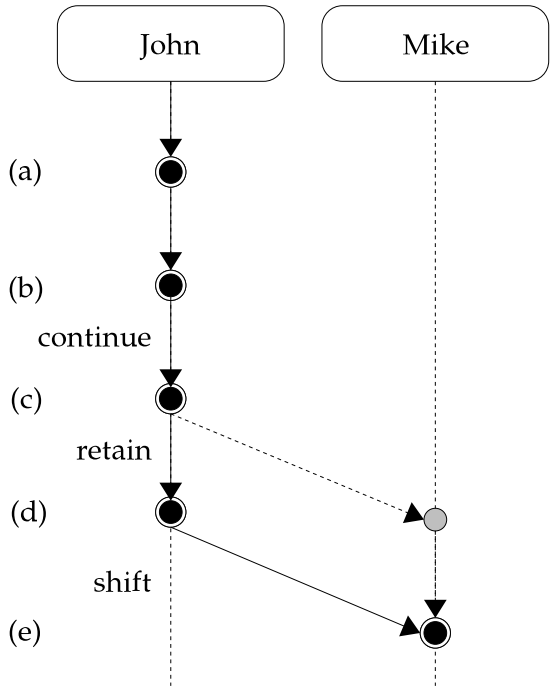


Figure 2.5: A sequence of centering transitions

vides a smoother transition of the center than the one in (20). Therefore Brennan et al. (1987) propose an extension of the inventory of translations which distinguishes between a "hard" shift and a "soft" shift.

Grosz et al. (1995) examine the linguistic features that are relevant for selecting an entity as the backward-looking feature of an utterance and find that the grammatical function of a forward-looking center has considerable influence on its likelihood to become the next backward-looking center, and furthermore, it is possible to rank the grammatical functions according to their influence on center selection.

First, they find that SUBJECT is ranked higher than PARALLELISM illustrated by the following example:

(22) a. Susan is a fine friend.
b. She gives people the most wonderful presents.
c. She just gave Betsy a wonderful bottle of wine.
d. She told her it was quite rare. (Susan told Betsy)
e. She knows a lot about wine. (Susan knows...)

(23) a. Susan is a fine friend.
b. She gives people the most wonderful presents.
c. She just gave Betsy a wonderful bottle of wine.
d. She told her it was quite rare. (Susan told Betsy)
e. Wine collecting gives her expertise that's fun to share. (Susan's expertise)

In both (d) utterances, *Susan*, realized in the subject position, is the backward-looking center. A preference of parallelism over subject position should result in different centers in the (e) utterances: While in (22-e), the pronoun remains in subject position, therefore leading to a preference of *Susan* as the backward-looking center, in (23-e) the pronoun in object position would realize the center *Betsy*. However, this is clearly not the case.

Grosz et al. (1995) find furthermore that there is a ranking of the influence of grammatical function:

SUBJECT > OBJECT(S) > OTHER COMPLEMENTS/ADJUNCTS

This finding is corroborated by other approaches that model anaphora, such as the rule-based Resolution of Anaphora Procedure by Lappin and Leass (1994), to be discussed in chapter 6, and by the feature selections of data-driven approaches, which show a clear influence and a ranking of grammatical function as well (chapter 8).

2.6 Discussion

We started this chapter with a basic distinction of two different variants of anaphora. *Sentence anaphora* occurs within the same sentence, and deals with the clearly complementary distribution of reflexive and reciprocal pronouns, non-reflexive pronouns and full NPs, which are determined by the rules of *binding theory*. We discussed three accounts of binding theory. The first one is the treatment of binding in GB-Theory (Chomsky, 1993), a syntactic configurational approach with the tree-relation of c-command at its core. In their HPSG-framework, Pollard and Sag (1994) present a syntactic non-configurational analysis which formulates the o-command relation based on a obliqueness hierarchy of grammatical functions. The analysis by Jackendoff (1972) finally is non-configurational as well, but the binding principles are expressed in semantic terms of a hierarchy of thematic roles.

We then moved on to Centering Theory (Grosz et al., 1995), which is a representative of the second variant of anaphora, called *discourse anaphora*. Essentially, it is an explicit formulation of factors or constraints that influence textual cohesion with a focus on coreference and anaphora. Its core assumption is that cohesion is related to processing load, and that proper structuring and proper signaling of the flow of centers of attention minimizes processing load and maximizes cohesion.

Both binding theory and Centering Theory have been used as the basis for formulating concrete algorithms for anaphora resolution. Brennan et al. (1987) present a procedural implementation of Centering Theory (see chapter 3). Beaver (2004) develops a declarative reformulation of Centering Theory within the framework of Optimality Theory.

In the original implementation of the rule-based Resolution of Anaphora Procedure for English by Lappin and Leass (1994), as well as the author's reimplementation of this approach for German (both chapter 6), a syntactic filter based on binding theory is employed to filter the candidate set of antecedents for a pronoun to be resolved. The filter implements three simple rules:

- Personal pronouns *must not* be contained in the candidate antecedent's argument domain.
- Personal pronouns *must not* be contained in the candidate antecedent's adjunct domain.
- Reflexive pronouns *must* be contained in the candidate antecedent's argument domain, and the candidate must fill a higher argument slot.

It is easy to see that the first two rules correspond to principle B of binding theory, while the third rule is a variant of principle A.[9] The difference to the original binding principles is that the rules do not make use of complex syntactic configurations: An NP is in the argument domain of another if they are siblings and arguments of the same verb – no deep syntactic analysis is required. Even without a notion of c-command and binding, the filter can handle the examples from the beginning of this chapter:

(24) a. John$_i$ hates himself$_i$.

[9] We do not consider full NPs in this thesis, so we do not need an implementation of principle C.

b. *Himself$_i$ hates John$_j$.
c. *John$_i$ hates him$_i$.

In (24-a), *himself* is in the argument domain of *John*, and *John* is the subject while *himself* is the object. Thus, the filter accepts the sentence. (24-b) is rejected because the reflexive pronoun is in a higher argument position than *John*.[10] The ungrammatical (24-c) is rejected by the first rule, stating that personal pronouns must not be in the argument domain of their antecedent.

All accounts that we described have in common that they use a ranked thematic or grammatical role hierarchy for restricting or suggesting the distribution of anaphora. The obliqueness hierarchy of grammatical roles (which is a syntactic phenomenon) is closely related to the semantic thematic role hierarchy – since thematic roles have a strong tendency to be regularly realized in specific syntactic argument positions. The fact that it is ultimately the thematic role hierarchy that obviously plays a central role in the description of anaphora mirrors the semantic nature of anaphora.[11] In the implementations that we are going to discuss in this thesis, this hierarchy is a vital source of information – both in rule-based approaches employing more sophisticated linguistic knowledge as well as data-driven ones which use just shallow information.

While so far, we focused on a rather descriptive view of the phenomenon of anaphora, in the next chapter, we will turn to issues that arise with the task of actually resolving pronouns. We will review data structures for the representation of coreference and anaphora as well as strategies for algorithmically solving anaphora on an abstract level. We will further discuss a number of concrete algorithms, which are to be considered prototypical examples of classes of algorithms, which later on in this dissertation will return in concrete implementations.

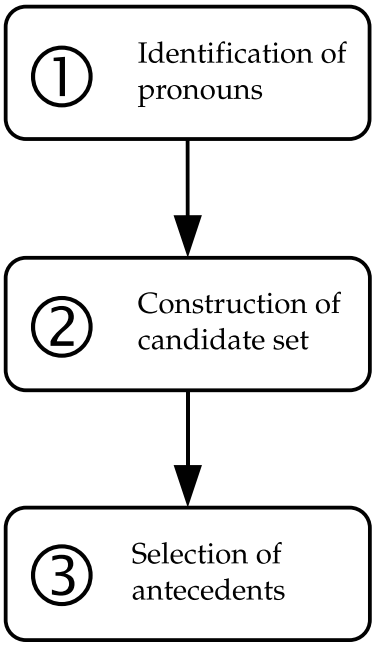
[10] Proper binding theory would reject (24-b) because *John$_i$* is bound by *himself$_i$* - a violation of principle C.

[11] Of course, the formulation of binding theory in terms of thematic roles is not without problems, as for example Pollard and Sag (1992) and Büring (2005) point out.

Chapter 3

Resolution Strategies

Numerous different approaches to tackling the problem of anaphora resolution have been suggested as the result of the research in this field. While these systems differ widely in they way they are implemented, what linguistic information they use and how this information is gathered, the sequence of basic abstract tasks that must be solved by a resolution system is the same for all the approaches. It can be outlined as follows:



Identification of pronouns Not all pronouns in a text do actually corefer with some other entity. The most frequent kind of such non-

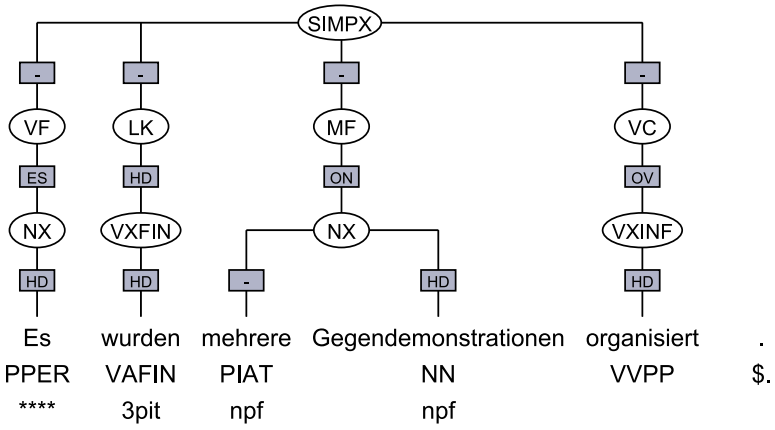


Figure 3.1: Expletive pronoun *es* in initial field position

referential pronouns are expletive pronouns, as in the following sentence:

(1) **Es** wurden mehrere Gegendemonstrationen organisiert.
There were several counter-demonstrations organized.
'Several counter-demonstrations were organized.'

Here, the expletive pronoun *es* fills the subject slot of the sentence in the initial field, while the actual subject *mehrere Gegendemonstrationen* has moved into the middle field (see figure 3.1).[1] The pronoun is not coreferent, and any antecedent that would be selected by the resolver would be an incorrect decision. Therefore, in the identification step, the resolver determines for each pronoun whether it is referential or not, and excludes all non-referential pronouns from further processing. It is reasonable to add this additional step prior to the pronoun resolution proper, since this way it is possible to fine-tune the implementations of the individual steps to their specific task. In particular, this way it is possible to assume in the actual resolution process that *every* pronoun must be resolved.

For English, several approaches to automatically detecting expletive pronouns have been suggested including pattern-based approaches (Paice and Husk, 1987), approaches employing machine learning

[1] This structure is called *Vorfeld-Es*.

methods (Evans, 2001; Müller, 2006), and systems that aim to combine both strategies (Boyd et al., 2005). The author's own resolution systems for German (to be discussed in chapters 6 and 8) rely on the manual gold annotation of expletives in the TüBa-D/Z corpus, which serves as the primary data source (see chapter 5).

Construction of candidate set For each pronoun to be resolved, a set of nominal elements (definite noun phrases or pronouns), called the *candidate set* is created in the second step. The candidate set contains all *potential* antecedents of the pronoun, among which the resolver must choose one or more elements that it considers correct antecedents. Frequently, filters are applied to the candidate set which reduce its size before it is passed on to the actual resolution step. An example of such a filter is the morphological prefilter in the author's systems (see section 8.1).

Selection of antecedents The core task of the resolver is to select one or more antecedents from the candidate sets.

Despite the great variety in their architecture, there is a limited number of general *resolution strategies* that are adopted essentially by all approaches, regardless of the specific way they are implemented, and what concrete linguistic information they use in order to arrive at their result, or the way this information is represented. The resolution strategies that we will identify in this chapter are located along three dimensions.

The first dimension concerns the *resolution model*. This is the way how the resolver processes the set of candidates in order to find one or more antecedents for a pronoun. In a *competition model*, multiple candidates are considered at the same time, and a ranking is imposed on them. Finally, the candidate that is ranked highest is selected as the antecedent. In *strictly pairwise models*, the decision whether a referential relation holds between a pronoun and a candidate solely depends on the information about the pair alone.

Resolution history is the second dimension and pertains to the dynamic nature of discourse. The salience of a referent in a discourse does not remain constant over time: The more frequently a referent is referred to, the more salient it is. In other words, with growing cardinality of the coreference set, the higher the salience of all of its members. There are also

dynamic factors that decrease salience, most importantly the distance between two mentions of a referent. In order to represent the dynamic change of salience, a resolution approach must have some means of modeling the real-time development in discourse. This is a property that is typically found in procedural approaches to anaphora which maintain an internal memory of referents already resolved, rather than approaches that treat each resolution of a pronoun to an antecedent as a separate task. Typically, data-driven methods based on machine learning do not maintain such an internal memory, as this would require on-the-fly re-computation of feature vectors of the instances to be classified. The author's rule-based approach for resolving German anaphora, to be discussed in section 6.3, does employ a representation of the dynamic change of salience, which is determined by the distance of the candidate to the pronoun and the candidate's mention count in the discourse.

The third dimension is the general kind of the *resolution algorithm* that is used to resolve pronouns. We will distinguish two types: *rule-based approaches* that make use of a set of explicit linguistically motivated rules on the one hand, and *data-driven approaches* that autonomously extract an implicit model from pre-annotated data.

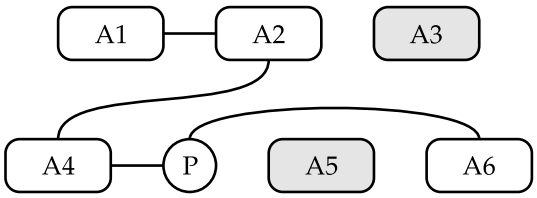


Figure 3.2: Schematic structure of a coreference chain

3.1 Representation of coreference

Figure 3.2 shows the schematic structure of a discourse, represented as a coreference chain. Seven entities are elements of this discourse: the noun phrase A1 through A6, and the pronoun P. A2 is coreferential to A1, A4 to A2, and A6 is coreferential to the pronoun P. P itself is anaphoric to A4. By the nature of a chain, an element of the chain is at most linked to two other elements. Only by taking the transitive closure, it becomes obvious that all elements of the chain actually refer to the same entity. A3 and A5 in figure 3.2 are not part of the coreference chain A1–A2–A4–P–A6 and refer to other entities.

The alternative mode of representation is a coreference set. This configuration is sketched in figure 3.3. The discourse elements that refer to the same referent belong to the same set.

Coreference chains and coreference sets are fully equivalent in what information they represent, but they are complementary in what information they represent *explicitly*: Coreference chains directly reflect the order in which the elements occur in a discourse by linking every element to its direct predecessor and successor. Given two arbitrary discourse elements however, the only way to find out whether they are in fact coreferent is to follow the chain starting at one of the discourse elements and test whether the other discourse element is a member of this chain as well. Coreference sets on the other hand represent the latter information explicitly. Here, it is the order of the discourse elements that can only be found out by indirect means, such as the position of the discourse elements in an uttered sentence, if this information is available.

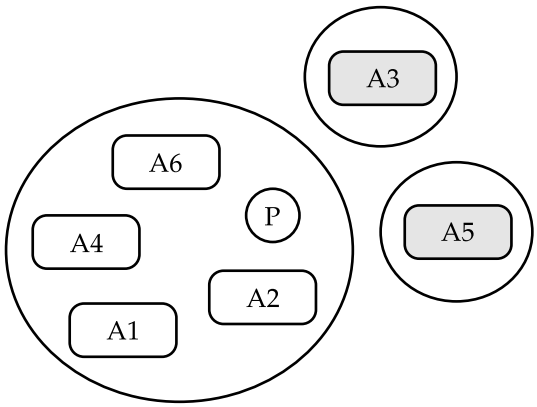


Figure 3.3: Schematic structure of a coreference set

3.2 Linguistic information

Any system for pronoun resolution must have access to the relevant linguistic information in order to perform its task. This section will concentrate on introducing the most important types of linguistic information that have been used in the literature. In the chapters that follow, specific kinds of linguistic information will be discussed in greater detail.

- **Positional information**: Positional information is information about the position of a candidate antecedent relative to a pronoun. Positional information can be used by resolution systems as an indicator of recency – more recent NPs are more likely to be antecedents – and to determine cataphoric relationships.
- **Syntactic information**: Syntactic information is used for example to represent syntactically expressed variants of binding theory. Reflexive pronouns are subject to strict binding conditions which are frequently sufficient by themselves to select a correct antecedent.

  Furthermore, information about the grammatical function of NPs can be used to model a measure of salience. Referents with higher salience are more likely to be antecedents.
- **Lexical information**: The lexical properties of two NPs are an important hint whether they are coreferent or anaphoric. For full NPs, many resolution systems take for instance a substring-match, i.e. the containment of the surface form of one NP in the surface form of the

other as evidence of their coreference. For the resolution of pronouns, lexical information is less significant as the variety of surface forms is obviously limited. However, even for pronouns, if the same pronoun occurs multiple times in a text, it is quite likely that this pronoun refers to the same entity.

- **Morphological information**: Agreement between a pronoun and a candidate antecedent can be a valuable indicator whether they are in a referential relation or not. A candidate antecedent that does not agree with the pronouns is less likely to be a correct antecedent than a candidate that does agree.
- **Semantic information**: Coreference is determined by semantics by considerable measures. Reliable information about the semantic properties of the involved referents is a valuable hint to what candidates can be correct antecedents.

3.3 Resolution models

At some point in any resolution process, regardless of the concrete design of the resolution system, a set of noun phrases is constructed that constitutes the set of the candidates. It is the core task of the resolution system to examine all the candidates in this set, and then to finally arrive at a decision which candidate to select as the antecedent for the pronoun. We distinguish two basic strategies of what kind of information a resolution system considers during the decision process, which we are going to call *resolution models*. In a *pairwise model* of resolution, only properties of the pronoun and the candidate that are members of the pair that is considered for resolution are taken into account. *Competition models* implement the resolution process as a competition between all the candidate antecedents of the pronoun.

3.3.1 Pairwise models

The prevalent characteristic of a pairwise model of resolution is that the decision whether a candidate pair is anaphoric or not is only made on the basis of information about the pronoun and the candidate that are members of the pair. The abstract structure of a pairwise resolution model is

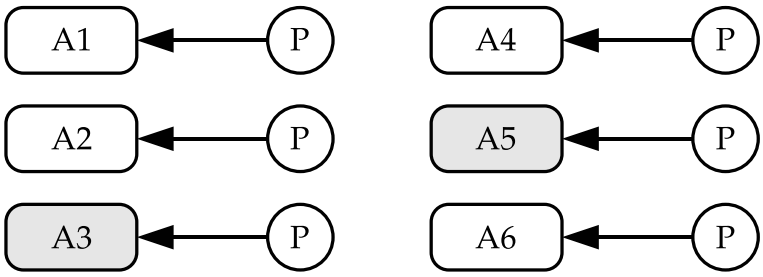


Figure 3.4: Pairwise model of pronoun resolution

illustrated in figure 3.4. The candidate set contains the elements A1-A6. Combining each of the candidates with the pronoun P yields six different pairs. The resolver then decides separately for each pair whether the pronoun is anaphoric to the candidate antecedent or not. The properties of the other candidates are not taken into account.

A side effect of the pairwise resolution approach is that it is possible or rather natural for a pronoun to be resolved to more than one antecedent. This is a consequence of the model's restriction to only pairwise resolution. Removing the remaining candidates from the candidate set would require a global view on the current status of the set and the resolution process which is not available in a pairwise resolution model.

3.3.2 Competition models

Competition models (for example Yang et al. (2003), Denis and Baldridge (2007) or Lappin and Leass (1994)) are the counterpart of pairwise models. While in a pairwise model, an antecedent is selected solely on information about the pronoun and one candidate at a time, the core strategy in a resolution system that is based on a competition model is to view the resolution process as a *competition* between multiple candidates. Specific to a competition model is the *ranking* of the candidates that is imposed by the resolver. To determine this ranking, the resolver looks at all candidates in the set and rates the candidates' properties according to an internal scale of importance. The candidate that is ranked highest is finally chosen as the antecedent.

Figure 3.5 illustrates the competition model. The candidate antecedents A1-A6 form a cohort of competing elements. Each candidate is assigned

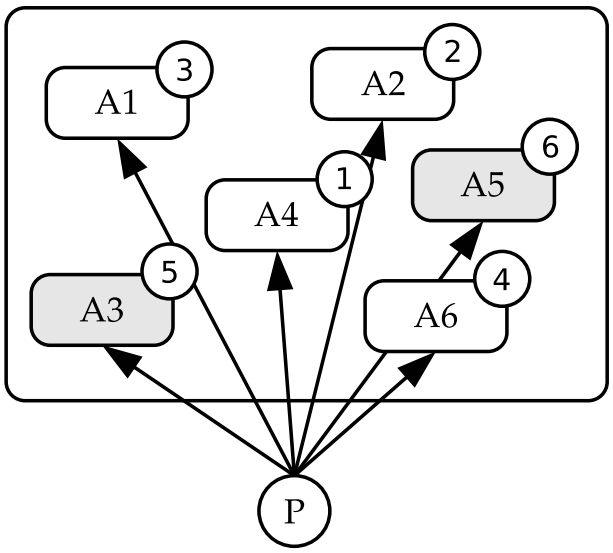


Figure 3.5: Competition model of pronoun resolution

a rank (depicted by the numbers in circles). The resolver considers the candidates in the cohort all at once and picks the one which turns out to have the highest rank. In the abstract example, A4 has the highest rank, therefore it is selected as the antecedent of P.

3.4 Resolution algorithms

The purpose of this section is to give an overview of the existing algorithms for pronoun resolution, and to classify the algorithms according to their underlying design principles.

In the literature on pronoun resolution, various computational approaches to solving the task of resolving a pronoun to the correct antecedent have been proposed. From a coarse perspective, these algorithms can be divided in two basic classes. Algorithms that belong to the first kind operate on the basis of a built-in set of predefined rules. Historically, they date back to the early 1970s. A more recent development are algorithms that belong to the second kind. They make their decisions based on classification principles that they acquire automatically using machine learning techniques, usually with pre-annotated corpora as training sets.

3.4.1 Rule-based approaches to pronoun resolution

Rule-based approaches to pronoun resolution can best be described as a set of constraints on a possibly very large set of candidate pairs of pronouns and potential antecedents. The constraints express the linguistic conditions that allow for a pair to remain in the set of candidates, and they directly relate to appropriate concepts in the linguistic theory, such as binding constraints or principles of silence of entities in a discourse. The constraints are designed manually by a linguist in the form of rules that are expressed explicitly in the system. This explicit representation of rules makes it possible to inspect the effect that any single rule has on the resolution process. By actually applying the algorithm and then evaluating its performance on a rule-by-rule basis, the utility of each rule can be measured. Further, since rules are closely tied to the underlying linguistic hypotheses, it is possible to judge their relative importance in the process of pronoun resolution.

While the characteristics of rule-based approaches outlined so far are certainly advantageous, there are also disadvantages: Formulating the necessary constraints and rules for the resolution algorithm requires a substantial amount of work on the part of the implementing linguist, with the consequence that it takes a long time until such a system is completed. Moreover, the rules created by the researcher might be subject to a certain linguistic bias in the sense that traditional insights about pronoun resolution that are evident in the consulted linguistic theories are over-emphasized in the rules while other interesting factors are overlooked. Hinrichs et al. (2005a) report that their memory-based resolution system rated a feature corresponding to the grammatical function FOPP in TüBa-D/Z *(optional PP complement)* among the six most important, more important than the ON (grammatical function subject) feature. The reason for this configuration is that NPs in optional PP complements are *highly unlikely* of being antecedents. Linguists with the appropriate theories in mind might tend to formulate rules involving "important" aspects such as the subject status of an NP - while overlooking factors beneficial for solving the task that are usually considered not central for pronoun resolution.

In the following section, we will discuss an early rule-based algorithm as prototypical example. Further rule-based approaches are listed in the taxonomy at the end of this chapter (table 3.1).

The resolution algorithm by Hobbs

One of the very first algorithms for anaphora resolution, and highly influential on the subsequent research in the field, is the rule-based algorithm by Hobbs (1978). Hobbs never implemented his algorithm on a running computer system. Instead, he applied the necessary steps by hand to 300 samples that he took from three texts of different genres. The algorithm assumes as its input fully parsed surface trees. Hobbs assumes a traditional $\overline{X}$-scheme for the underlying grammar.

In order to find antecedents for a pronoun, the Hobbs algorithm traverses the surface parse trees in eight steps, as outlined in the following.

1. *Begin at the NP node immediately dominating the pronoun.*

   The algorithm works its way left-to-right through the parse trees, proposing an antecedent for each pronoun it encounters.

2. *Go up the tree to the first NP or S node encountered. Call this node X, and call the path used to reach it p.*

   The algorithm inspects nodes that are closer to the pronoun first.

3. *Traverse all branches below node X to the left of path p in a left-to-right, breadth-first fashion. Propose as the antecedent any NP node that is encountered which has an NP or S node between it and X.*

   This step serves two purposes:

   - By restricting the search only to nodes to the left of the path p that leads from the pronoun to X, the algorithm excludes NPs that follow the pronoun from the list of possible antecedents.
   - The requirement that an NP or S nodes occurs between the candidate NP and X makes sure that for a non-reflexive pronoun, no antecedents are found that are located in the same simplex sentence, such as

     (2) *John$_i$ likes him$_i$.

4. *If node X is the highest S node in the sentence, traverse the surface parse trees of previous sentences in the text in order of recency, the most recent*

first; each tree is traversed in a left-to-right, breadth-first manner, and when an NP node is encountered, it is proposed as antecedent.

The algorithm reaches the root node of a parse tree when there are no more candidate antecedents to be found in the current sentence. In this case, it successively searches for more candidates in the preceding sentences. Unlike for candidates in the current sentence, the algorithm does not impose any syntactic constraints on candidates that are located in previous sentences.

If X is not the highest S node in the sentence, continue to step 5.

5. *From node X, go up the tree to the first NP or S node encountered. Call this new node X, and call the path traversed to reach it p.*

6. *Traverse all branches below node X to the* left *of path p in a left-to-right, breadth-first manner. Propose any NP node encountered as the antecedent.*[2]

7. *If X is an S node, traverse all branches of node X to the* right *of path p in a left-to-right, breadth-first manner, but do not go below any NP or S node encountered. Propose any NP node encountered as the antecedent.*

8. *Go to step 4.*

To illustrate the functionality of the algorithm, Hobbs (1978) gives an example, which is repeated here:

(3) The castle in Camelot remained the residence of the king until 536 when he moved it to London.

This sentence is parsed into the parse tree shown in figure 3.6.

In order to find antecedents for the pronoun *it*, the algorithm starts at NP_1. Step 2 selects S_1 for X. The path p now contains the nodes it–NP_1–VP–S_1. Step 3 searches the tree to the left of p, but no candidate antecedents can be found. Step 4 does not apply, since S_1 is not the root node of the tree. Step 5 moves up to NP_2. Step 6 proposes the head of NP_2 as a candidate antecedent, which is *536*. Steps 7 and 8 do not yield any additional candidates. Step 9 returns back to step 4. Step 4 does not apply. Step 5 rises to S_2 and then the algorithm skips step 6, which only applies

[2] This should rather read "propose *the head* of any NP node encountered as the antecedent".

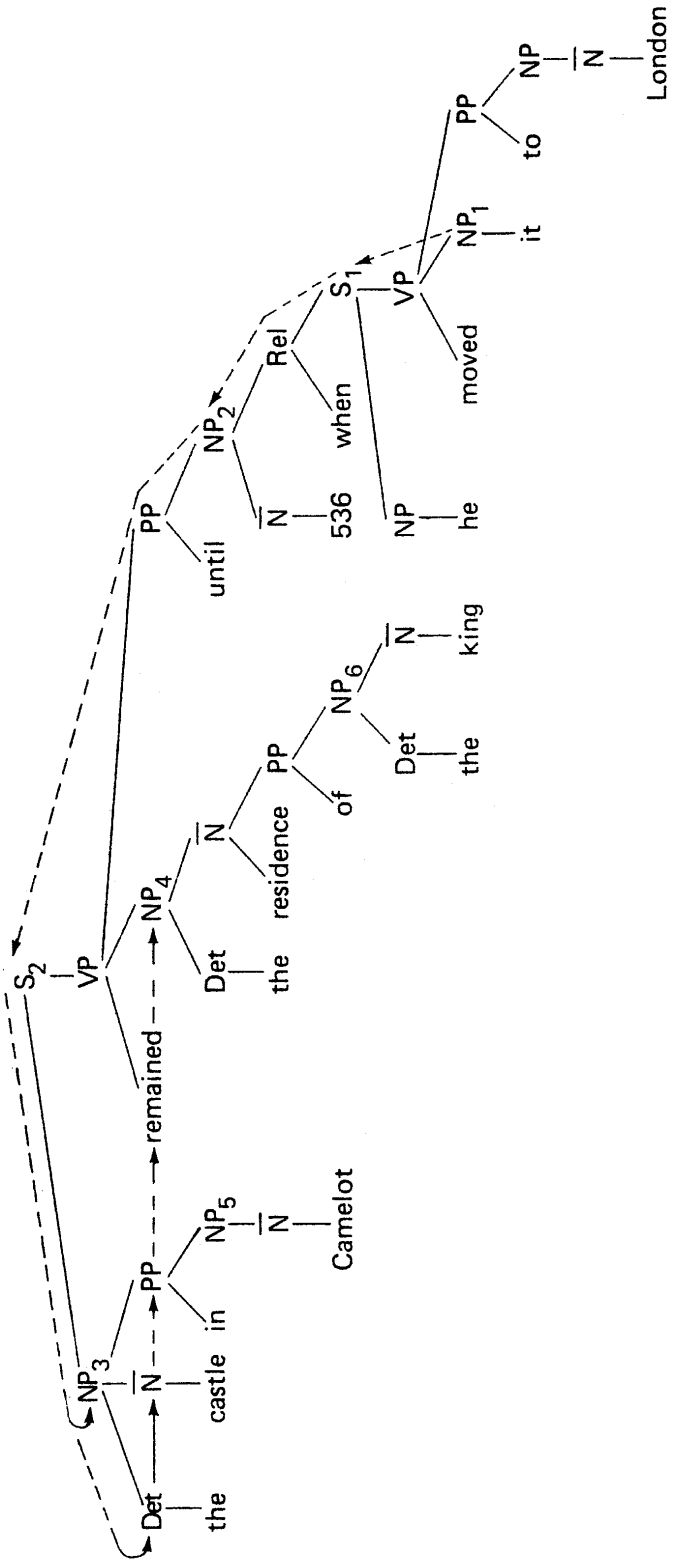


Figure 3.6: Progress of Hobbs' algorithm.

to NP nodes. In step 7, two NP nodes are suggested: NP_3, *castle*, and NP_4, *residence*.

Thus, the algorithm ends up with a set of three suggestions: *536*, *castle*, and *residence*. There is no way to rule out the incorrect candidates on purely syntactic grounds. For this reason, Hobbs retreats to the semantic level. He points out that the selectional preferences of the verb *to move*, that the pronoun in question is the direct object of, require that the object be movable. However, dates can't move, which rules out *536*. The same argument applies to *castle*, which is rejected as well, leaving *residence* as the only and correct candidate.

It should be noted at this point that the fact that Hobbs' algorithm must consider semantic properties of the verb in order to arrive at the correct antecedent suggests that purely syntactic constraints may not be sufficient to solve the task. For an algorithm that is executed by hand this is not much of an issue, as the executing linguist has access to arbitrary, or at least sufficient, lexical and world knowledge. However, this poses a serious problem for algorithms that are designed to be actually implemented on a computer, since the resources available here are much more limited.

In chapter 6 we are going to discuss in detail a rule-based resolution algorithm by Lappin and Leass (1994) which was actually implemented as a running program. Unlike Hobbs' algorithm, it does not rely on a tree-walk to find candidates and determine syntactic configurations that restrict what can be an antecedent. Instead, it uses a ranked grammatical role hierarchy to model the salience of discourse referents and a shallow syntactic implementation of binding principles to determine antecedents. By combining a salience measure and syntactic binding restrictions, the algorithm applies to both discourse anaphora and sentence anaphora. The author's rule-based anaphora resolution system for German, which will also be covered in chapter 6, is modeled after this system.

The resolution algorithm based on Centering Theory by Brennan et al.

Brennan et al. (1987) present an algorithm for pronoun resolution based on Centering Theory. It is embedded as a module in the authors' implementation of an HPSG natural language system and interacts with other modules such as a syntactic analyzer, a resolution module for sentence anaphora, or a module for determining semantic concepts. Thus, the algorithm is a knowledge-rich approach. The algorithm operates in three steps, which are (i) the construction of anchors,[3] (ii) filtering of the proposed anchors and finally (iii) the classification and ranking of the remaining anchors.

In the construction step, the lists of forward looking centers $C_f(U_n)$, and a list of possible backward looking centers $C_b(U_n)$ are created for utterance U_n. The forward looking centers consist of all referring expressions, ordered by grammatical role. For pronouns, one entry is created for each possible agreeing discourse entity, for proper nouns an entry is added for each possible referent. For the backward looking centers, the list of forward looking centers from the previous utterance is re-used. Each element of the C_b list is paired with the C_f, yielding a set of possible anchors.

In the filtering step, entries for pronouns are removed which are indexed with multiple disjoint discourse entities at the same time. Furthermore all anchors are removed where the C_b is not equal to the first element on the C_f list (recall that Centering Theory requires the C_b to be the most likely $C_f(U_{n-1})$). Finally, all anchors are eliminated that do not meet the requirement that when one of the C_f is realized as a pronoun, the C_b must also be realized as a pronoun.

The ranking step finally applies the ranked translation rules to the remaining anchors. The anchor which is assigned the highest rank is selected.

The centering algorithm is an example for a genuinely discourse-oriented resolution approach. A variant of centering is presented by Strube (1998), who reduces the lists of centers to just one list of referents ordered by salience.

3.4.2 Data-driven approaches to pronoun resolution

In addition to the fact that the creation of a comprehensive rule set is an extremely time-consuming task, it can only be developed by a long-time

[3] An anchor is a pair $\langle C_b(U_n), C_f(U_n)\rangle$.

expert in the field. The reason for this is that rules should cover as many aspects of the relevant phenomenon as possible. Formulating such a rule set requires substantial expertise.

Data-driven methods take a very different approach. At their core, a machine learning algorithm is employed that autonomously computes an internal model of the phenomenon in question, for example anaphoric relations between pronouns and antecedents, from a large set of samples that are extracted from pre-labeled data. Thus, it is not necessary to formulate any rules, as they are determined by the machine learning system itself in a black-box manner. The expert knowledge is of course in the data, which must be pre-labeled by hand.[4] An example would be the German TüBa-D/Z newspaper corpus, which contains manual annotations for anaphoric relations between pronouns and antecedents, among others (see chapter 5). Of course, the manual annotation of raw data is also very time consuming, and requires substantial knowledge. The difference to the task of designing a rule system is that the annotation of data requires the application of a pre-determined rule system to a series of *concrete* and *independent* examples. It is not necessary to be aware of all the relevant cases at the same time as when developing a rule system. Thus, annotation of data can also be accomplished with satisfying quality by annotators who do not have long-standing expertise. Furthermore, due to the fact that the annotation decisions are independent of each other, the work can be easily distributed among multiple annotators.

Data-driven approaches usually operate in two stages. The first stage is the *training stage*, in which the machine learning system builds its internal model from a set of training samples that were extracted from the annotated data. Before they can be used, the training samples must be transformed into a format that the machine learning system understands. For many machine learning systems, this is a set of *feature vectors* which represent properties considered relevant for the task. For pronoun resolution, features frequently used are the distance between the pronoun and the antecedent, the grammatical role, morphological agreement, and so on – the linguistic knowledge as explained in section 3.2. The second stage in the *classification stage* in which the machine learner applies the internal model

[4] Machine learning approaches that employ manually pre-labeled data are called *supervised* approaches.

that it acquired during the training stage to new samples which must also be represented as feature vectors.

We will discuss a number of resolution systems based on machine learning in this thesis, among them a system based on a decision tree classifier by Soon et al. (2001) and a system based on memory-based learning (Preiss, 2002) in chapter 7. The author's own hybrid resolution system (see chapter 8) uses a memory-based resolution component at its core. The following section includes references to further resolution systems, based on rules as well as on data-driven approaches.

3.5 A taxonomy of resolution algorithms

In the previous section we characterized the properties of approaches to anaphora resolution on an abstract level. This section gives an overview of existing implementations and categorizes them along the dimensions introduced so far. We restrict ourselves to approaches that focus on pronoun resolution or that equally consider pronoun resolution and the resolution of full NPs. We do not discuss work that focuses on the resolution of full NPs, as these approaches usually employ traditional pronoun resolution techniques. The reader is referred to chapter 9, which provides a compact survey of some of the work in this area.

Table 3.1 shows a classification of several approaches to pronoun resolution according to the following properties:

rule-based: the resolution approach uses a rule-based algorithm

data-driven: the resolution approach makes use of data-driven (machine learning) methods

dynamic: the algorithm models the dynamic change of salience of discourse referents

positional: the algorithm considers positional features, such as the distance between pronoun and antecedent

syntactic: the algorithm considers syntactic features, such as the grammatical function of an NP or syntactic representation of binding principles

lexical: the algorithm considers lexical features, such as a substring match between two NPs

morphological: the algorithm considers morphological features, such as agreement in number and gender

semantic: the algorithm considers semantic features, such as the semantic class of an NP (usually by lookup in WordNet)

salience hierarchy: the algorithm assigns salience values on the basis of a ranked grammatical role hierarchy (this is actually an instance of a syntactic feature)

procedural: the algorithm is described in a procedural, i.e. step-by-step framework that may mirror the real-time flow of a discourse

declarative: the algorithm is formulated declaratively, usually by a set of constraints which are applied by constraint solver[5]

classifier: the classifier used, for approaches based on machine learning

combined classifiers: multiple classifiers work synchronously to obtain the final result (such as co-training or re-ranking)

hybrid: hybrid approaches combine different architectures, such as rule-based pre- or postfiltering, and machine-learning-based resolution

competition: the resolution approach implements a competition model of pronoun resolution

language: the target language, where EN is English, DE is German, and JP is Japanese.

[5]Note that machine-learning-based approaches do not fit in the procedural/declarative pattern. They do not model a time sequence of resolution decisions, but they do not use sets of constraints either. Their processing model is a sequence of atomic resolution decisions.

Approach	rule-based	data-driven	dyn-amic	posi-tional	syn-tactic	lexi-cal	morpho-logical	sem-antic	sal. hier.	proce-dural	declar-ative	clas-sifier	comb. class.	hybrid	compe-tition	lang.
Aone and Bennett (1995)		×		×	×			×	×			C4.5				JP
Azzam (1996)	×		×							×					×	EN
Beaver (2004)	×				×						×				×	
Dagan and Itai (1990)		×			×					×						EN
Denis and Baldridge (2007)		×		×	×	×	×					MaxEnt			×	EN
Ge et al. (1998)	×	×		×	×		×	×						×		EN
Hobbs (1978) *	×				×			×		×						EN
Kehler (1997)		×		×	×							MaxEnt				EN
Kennedy and Boguraev (1996) *	×			×	×				×	×						EN
Lappin and Leass (1994) *	×		×	×	×				×	×					×	EN
McCarthy and Lehnert (1995)		×		×		×		×				C4.5				EN
Mitkov (1998)	×				×					×						EN
Müller et al. (2001)		×		×	×	×	×	×	×			J48	×			DE
Ng (2005)		×		×	×	×	×	×				C4.5/RIPPER/ MaxEnt	×			EN
Ng and Cardie (2002)		×		×	×	×	×	×				C5		×		EN
Preiss (2002) *	×	×		×	×				×	×		TiMBL		×		EN
Soon et al. (2001) *		×		×	×	×	×	×				C5				EN
Stuckardt (2001)	×		×	×	×	×	×		×	×						EN
Stuckardt (2002)		×										C4.5				EN
Yang et al. (2003) *		×		×	×	×	×					C5			×	EN
Yang et al. (2005)		×			×			×				C5			×	EN
Wunsch (RAP-G) *	×		×	×	×		×		×	×					×	DE
Wunsch (hybrid) *	×	×		×	×		×	×	×			TiMBL		×		DE

Table 3.1: Pronoun resolution systems. Entries marked with * are discussed in more detail in this thesis.

Chapter 4

Evaluation Strategies

Key to any system for coreference resolution, but frequently overlooked in its importance, is the evaluation step that scores the results generated by the system. Only with meaningful scores, it is possible to evaluate the system's performance and to compare it to the performance of other systems. In order to be useful, a scorer should meet a number of basic properties:

- **Applicability** of the scorer to the data
- Only the **relevant** information should be scored
- The scores should match our **intuitions** about correctness

Applicability concerns the preconditions that must hold in the data in order for the scoring approach to produce valid results. Scoring approaches are not independent of the information contained in and the structure of the data to be scored. If these implicit assumptions are not met in the data, the results will be invalid. The model-theoretic coreference scoring approach to be presented in section 4.4 for instance assumes that all coreference relations to be scored are equivalence relations. However, there exist coreference relations that are not equivalence relations, to which the approach cannot be applied.

Relevance means that a scoring approach should capture those aspects of the output that are relevant for properly evaluating a system. For example, in pronoun resolution the vast majority of possible pairings of pronouns and antecedents are *not* in any referential relation. The resolver makes a correct decision if it does *not* annotate the pair. Whether these

negative samples might be taken as relevant or not depends on the problem at hand. The distribution over the instances is extremely skewed towards the negative instances – they outweigh the positive instances by several orders of magnitude. If the negative instances are not counted, a very large number of correct decisions will be ignored, since most of the cases are negative instances. If the negative instances are counted, the number of positive instances will become negligible compared to the number of negative instances. This may be undesirable, as the more interesting insights might more frequently be found with the positive instances.

The **intuitiveness** of a scoring approach captures to what extent it matches humans' intuitions about correctness. Under certain circumstances for example, scorers tend to over-penalize incorrect decisions or even misinterpret correct decisions as being wrong (usually for the reason that one or both of the other properties are not met). One such case is discussed in section 4.4.

In the remainder of this chapter, we will first introduce several mathematical scoring measures that are relevant to coreference resolution and characterize them with respect to the properties discussed above. Further, we will discuss two different data models over which scores can be calculated, the first of which being based directly on the referential links established by a scorer, and the second adopting a model-theoretic view of referential relations for scoring.

4.1 Precision and recall

Precision and recall are scoring measures that are widely used throughout all disciplines that require numerical scores of classifications against a gold standard, among them many fields related to computational linguistics, such as parsing (Black et al., 1991), information extraction, and anaphora resolution. On a somewhat abstract level, it is common to all of these systems that they perform a *search* through a space of possible objects with the goal of finding the ones relevant for the task. Objects of relevance are constituents in parsing, and structured chunks of information in information extraction. In anaphora resolution, the referential relations between the entities in a discourse are relevant.

As the result of its search, the system returns a set of objects that it

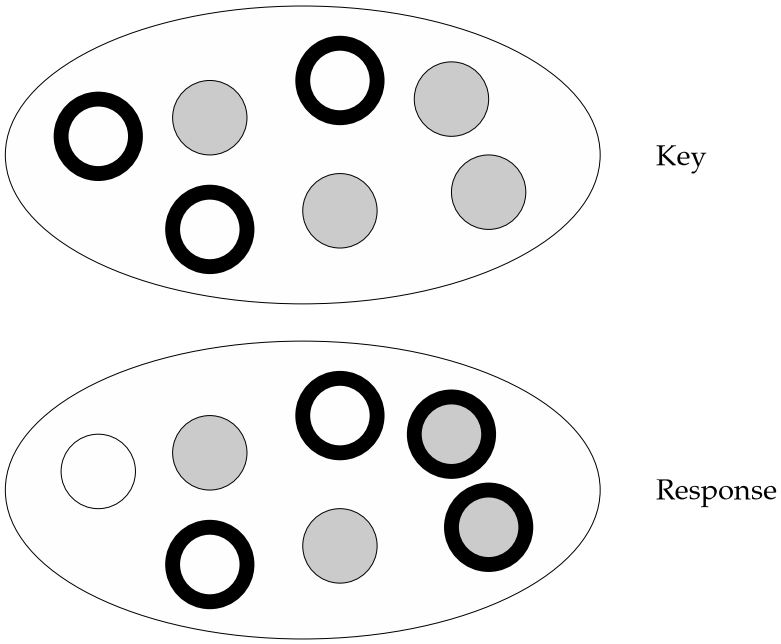


Figure 4.1: A key and a partly incorrect response. The task is to find white circles. The classifier failed to find one white circle, but erroneously selected two gray ones instead.

considers relevant. This set of objects is called the *response*. For evaluation, the response is compared to a pre-determined correct set of objects, which is called the *key*.

In figure 4.1, a key and a response are illustrated. The set of objects consists of white and gray circles. Assume that the task of the system would be to select from the set all white circles. The key encodes the correct solution to this problem, indicated by the thick lines around the white circles. For some reason the system does not work 100% accurately (usually, the task is much more complex than just to select white circles). It comes up with the response shown in the lower ellipse in figure 4.1. The system selected two white circles. It omitted one white circle, but selected two gray circles instead.

Two pieces of information are of relevance with respect to the response. The first one concerns the question how many of the relevant objects the system was able to find at all. This is called *recall*, and is defined as the ratio of the relevant objects in the response and the relevant objects in the

key. In figure 4.1, there are three relevant objects in the key – three white circles, but only two of them are also selected in the response. The gray circles were not part of the task, and thus they are not relevant. So the value for recall in this example is $\frac{2}{3}$. Recall can be computed as follows:

$$\text{Recall} := \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

The *true positives* are the correct objects in the response, that is, the selected white circles in the example. The *false negatives* are those objects that would have been relevant, but are missing from the response because the system made wrong decisions.

The second quantity is *precision*. It measures how many of the objects that the system found are actually relevant. Precision is defined as the ratio of the *actually* relevant objects (that is, the true positives) and the objects that the system *considered* relevant. In figure 4.1, the objects that the system considered relevant are all circles with thick lines in the response, that is 4. This is the sum of the true positives and the *false positives*, those objects that the system erred when selecting them. So for the example, we get a precision value of $\frac{2}{4} = \frac{1}{2}$. Thus, the formula for precision is:

$$\text{Precision} := \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Precision and recall measure two complimentary properties of the system. On the one hand, a system can be tuned to deliver results as correct as possible. This means that it must implement a very rigid search procedure, with the result that the system is biased towards dropping an object when it is not sufficiently confident even though the object would have been correct. A system tuned this way will maximize precision (in chapter 8, we will return to this question in more detail).

On the other hand, one might want to get as many objects as possible. This will result in more objects to be included in the response, increasing recall. However, some of the additional objects may be wrong. Thus, when precision is increased, recall drops. If recall is increased, precision drops. In the example in figure 4.1, precision is $\frac{1}{2}$, and recall is $\frac{2}{3}$, thus recall is higher than precision. This means that this example system aims to find as many as many objects as possible, which comes at the price that more incorrect objects are selected as well.

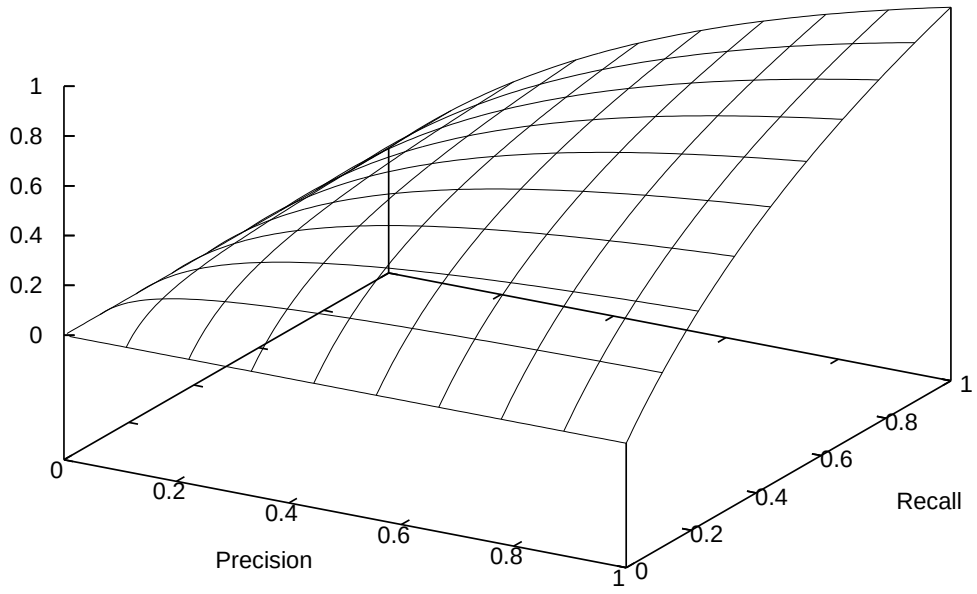


Figure 4.2: The f-measure

When designing a system, usually a trade-off between both precision and recall is aimed at. A measure that gives a notion of the combined performance of a system is the *f-measure*, which is a value related to the harmonic mean of precision and recall. Figure 4.2 illustrates it. It shows that the f-measure (depicted by the surface) continuously rises with precision and recall, reaching its maximum with the maximum of the two latter measures. This meets our intuition that the combined f-measure should reach its maximum when both precision and recall are maximal. Furthermore, the influence of precision and recall is evenly distributed as desired, and ensured by the mathematical properties of the harmonic mean.

The f-measure is calculated as follows:

$$F := \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

with P and R the values of precision and recall, respectively, and a weighting factor α. Usually, precision and recall are equally weighted by setting $\alpha = 0.5$. Then, the f-measure simplifies to

$$F := \frac{2PR}{(R+P)}$$

which is the form usually mentioned in the literature.

4.2 Link based and class based scoring schemes

4.2.1 Link based scoring schemes

The most obvious approach for assessing the performance of a coreference resolution system is to directly evaluate the links it created. To this end, scorers usually rely on the chain representation of the coreference relation (as discussed in chapter 3), where the links between two coreferent entities are explicitly represented. When evaluating the system, the scorer compares the links in the response (which were generated by system) to the links in the key.

The results are usually stated by means of precision and recall. The following configurations of links present or missing in the key or the response are possible, each of which with specific influence on precision and recall:

- **Identical links in key and response**
  This is counted as a true positive.
- **Link in the response points to the wrong antecedent**
  This is counted as a false positive, and is a precision error.
- **Link in the response that is not present in the key**
  This is a precision error as well, and counted as a false positive.[1]
- **Link that is present in the key, but missing from the response**
  This is a recall error, and counted as a false negative.
- **No link in the key, no link in the response**
  Two entities that are not coreferent are not linked in the key. If the resolver decides not to link the two entities in the response either, this is a correct decision. Pairs that are neither linked in the key nor in the response are called *true negatives*.

4.2.2 Class based scoring schemes

Class based scoring schemes are an extension to purely link based scoring schemes. Link based scoring schemes only examine the direction and an-

[1] The difference of this third case to the second case is that here, no link at all is going out from a pronoun (which may be an expletive pronoun, for example), while in the second, there *is* a relation between the pronoun and an antecedent in the key, but a different one than in the response.

chor points of two links in the key and the response. Thus, the links D – A and D – B in the response in figure 4.3 on page 69 will be counted as false positives, because they do not occur in the key. However, under the coreference class interpretation, the links are correct: They point to an antecedent that is in the same coreference class as the antecedents in the key. Class based scoring schemes consider a link correct if it connects two entities that are in the same coreference class. This yields the following slightly relaxed set of possible configurations:

- **Link connects two entities that are members of the same coreference class**
  This is counted as a true positive.
- **Link connects two entities that are members of different coreference classes**
  This is counted as a false positive, and is a precision error.
- **Link in the response that is not present in the key**
  This is a precision error as well, and counted as a false positive.
- **Link that is present in the key, but missing from the response**
  This is a recall error, and counted as a false negative.
- **No link in the key, no link in the response**
  This is a true negative.

4.2.3 Discussion

At the beginning of this chapter three important properties of scoring approaches were introduced: applicability of the approach to the data to be evaluated, relevance of what is evaluated, and intuitiveness of the scoring results. It is obvious that link based approaches can only be applied to data where referential relations are represented by explicit links. For class based scoring approaches information about the membership in coreference classes of the involved entities is necessary.

Both link and class based approaches are *binary scoring approaches*. This means that all individual scoring operations depend on only two entities and the link connecting them. An alternative approach (which we are going to discuss in section 4.4) would be to look at a complete coreference chain or

set *as a whole* and evaluate whether all elements that should be a member of that chain or set are in fact members. A binary scoring approach produces meaningful results in particular even if the response does not contain full coreference chains. For this reason, we chose to deploy a binary class based scoring approach for evaluating the resolution systems to be presented later in this thesis, as these systems are restricted to the resolution of pronouns only, i.e. coreference between definite NPs is not addressed.

With respect to relevance, the most important question is how to handle true negatives. As elaborated on above, true negatives are pairs of entities that are not in a referential relation, and correctly classified as such by the resolution system. As apparent from the formulae of precision and recall, true negatives are not considered at all in the calculation of these measures. With respect to their frequency of occurrence, the true negatives are by far the most numerous, and – since they represent correct decisions – it may be desirable to include them in the evaluation. In this case, they would be counted as true positives instead of true negatives. Depending on the perspective, this might seem a fairer approach than to ignore the true negatives as a whole. However, the distribution of negative instances and positive instances is extremely skewed towards the negative instances. Thus, when considered for evaluation, just by their sheer frequency, the true negatives will render all other instances completely insignificant. Considering that the more interesting and challenging cases are typically to be found among the positive instances, it might therefore be advantageous not to include true negatives in the evaluation. This is the strategy that we chose for the scoring system used in this dissertation.

A disadvantage of link based scoring approaches and, to a lesser degree, class based approaches, is that they tend to return results that are sometimes unintuitive to the human interpreter. The work by Vilain et al. (1995), to be introduced in section 4.4, is dedicated to this peculiarity and proposes an alternative approach.

4.3 Success rate

In the previous section, we defined recall to be the ratio of true positives, the number of correctly classified instances, and the sum of the true positives and false negatives, which is the number of instances in the key that the classifier should have found:

$$Recall := \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.1}$$

Mitkov (2002) points out that in the literature, two ways of defining recall have been proposed. The first variant is due to Aone and Bennett (1995), who define the measure as follows:

$$Recall_{ab} := \frac{\text{Number of correct anaphors}}{\text{Number of anaphors found by the program}} \tag{4.2}$$

The number of anaphors found by the program is the sum of the true positives in the sense above, i.e. the number of anaphors correctly classified, and the anaphors that the program found but that are not correct – i.e. the false positives. Thus, unlike definition 4.1, Aone and Bennett's version of recall does not relate to the key, but only to the response. Mitkov criticizes that defining recall this way has two major disadvantages: Firstly, for a system that is robust, i.e. selects an antecedent for every anaphor, recall defined this way is indistinguishable from precision. Secondly, it would be possible for a system to artificially raise recall to high levels just by resolving a very small number of very easy antecedents. In the extreme, the resolution of only one single trivial element to the correct antecedent would lead to a recall of 100%.

The definition of Baldwin (1997) is equivalent to the definition in 4.1:

$$Recall_b := \frac{\text{Number of correct anaphors}}{\text{Number of all anaphors}} \tag{4.3}$$

This variant of recall relates the number of correct anaphors (found by the program) to *all* anaphors *in the gold standard*, not only those found *by the program*. This much more meets the intuition of recall as described in section 4.1, and makes sure that the figures given by the measure are significant indicators of the success of the resolution process.

To avoid this inconsistency in the definition of recall, Mitkov introduces a third measure, called *success rate*, which he defines as follows:

$$\textit{Success Rate} := \frac{\text{Number of successfully resolved anaphors}}{\text{Number of all anaphors}}$$

Success rate relates the number of successfully resolved anaphors (i.e. the number of correctly resolved anaphors) and the number of all anaphors in the key. The results of the basic version of success rate are therefore equal to those given by recall. However, Mitkov additionally defines two "flavors" of the basic success rate measure that capture the graded increasing difficulty of an individual resolution decision: **Non-trivial success rate** evaluates only those anaphors that have more than one potential antecedent, and ignores those anaphors for which only one candidate antecedent exists. For some anaphors, even after applying gender and number filters, the set of candidates contains still more than one element. **Critical success rate** captures only those anaphors. Mitkov (2002) argues that it is the critical success rate that is indicative of the performance of a resolution system, and that it subsumes the less strict evaluation models.

4.4 A model-theoretic scoring scheme

For the coreference task of the sixth Message Understanding Conference (MUC-6), Vilain et al. (1995) suggested an alternative scoring scheme that is based on a model theoretic view of coreference rather than on the consideration of the actual coreference links. The reason for this choice is that link-based scoring schemes tend to give evaluation results that do not match our intuition of the severity of an error, when the gold standard and the annotations to be evaluated differ.

The following example illustrates this. Assume four entities that are coreferent, such as in the upper row in figure 4.3. The upper row represents the gold standard annotation, which links each entity to its direct antecedent, thus forming a coreference chain. The lower row represents the annotation to be scored. This might be an annotation that was created by a computerized coreference resolution system, or a second annotation, produced manually, which some measure of inter-annotator-agreement is to be computed for. At first glance, the two annotations are quite different. No coreference chain is recognizable and moreover, the links from D to C and from C to B are missing.

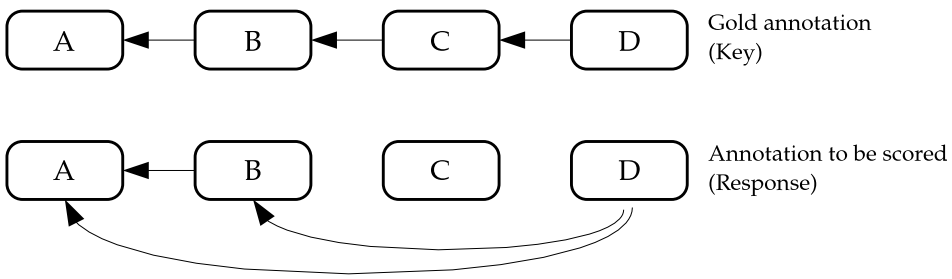


Figure 4.3: Different key and response for coreference relations

However, looking more closely, there is not at all that much difference as it seems. Recalling from chapter 3, there are two possible representations of referential relations, which are fully equivalent: The first one is a coreference chain, as in the top row in figure 4.3. The second way of representation is a coreference set, as shown in figure 4.4. The upper box depicts the coreference set of the key. All entities are members of the same coreference set - which is equivalent to all elements being linked by the same chain. The lower box represents the response. Entity C ended up in its own coreference set, as it is not linked to any other entity in figure 4.3. But the three other entities A, B, and D remain in the same coreference set. This is because the coreference relation partitions the entities in the discourse in *equivalence classes* – with all coreferent entities being added to the same coreference class. It does not matter exactly which two entities are linked, as long as in the transitive closure of all linked entities none of the coreferent ones is missing.[2] Thus, what looks as a severe difference of annotation in the chain representation in figure 4.3 turns out to be not that much of a difference when considering the coreference set representation.

If we calculate precision and recall figures based on the links in figure 4.3, we get the following: Of the three links in the response, only one link is present in the key: the link between B and A. Thus, we get for precision:

$$Prec = \frac{1 \text{ correct}}{3 \text{ total}} = \frac{1}{3}$$

For recall, there are three links in the key which should be present in the response, but only one was found:

[2]Of course, the notion of a linear order is lost when entities are linked differently, but the linear order can be restored from the natural linear order in the text. See chapter 3.

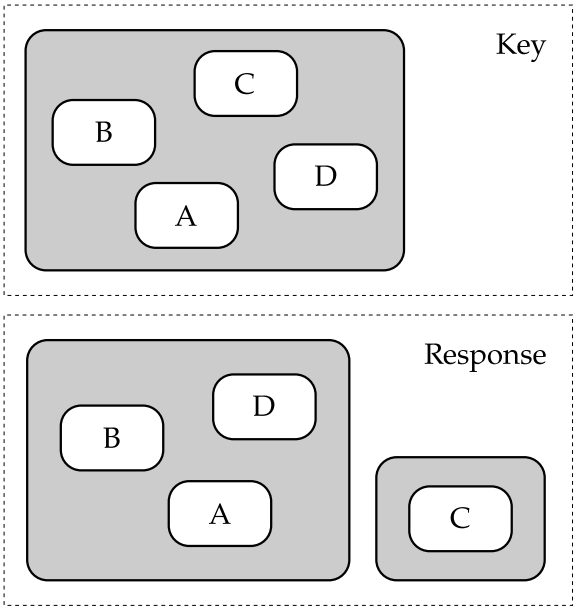


Figure 4.4: Coreference set view of the example

$$Rec = \frac{1 \text{ correct}}{3 \text{ total}} = \frac{1}{3}$$

$\frac{1}{3}$ for both measures is a very low value, considering that the actual error is only that the entity C is missing from the coreference set. The reason for this very unintuitive result is that this approach of evaluation ignores that with respect to the equivalence class, two links might have the same meaning, even though they are not identical.

Instead of directly evaluating links, the scoring approach of Vilain et al. (1995) considers the equivalence classes of the coreference relation aiming to arrive at more intuitive scores.[3] The basic idea is that a minimal spanning tree is created that connects the elements of the equivalence class formed by the key and the response. The difference that is measured is the number of links that must be added to the minimum spanning tree in order to arrive

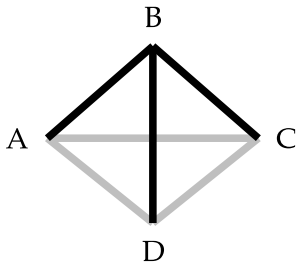
[3]It must be pointed out here that this approach is only valid for real equivalence relations. The *coreferential* relation and the *anaphoric* relation, as defined in the TüBa-D/Z annotation scheme for referential relations (see chapter 5), are true equivalence relations: If these relations hold between two markables, then they refer to *the same* extralinguistic entity, and the two markables therefore form an equivalence class.

at equal equivalence classes for the key and the response.

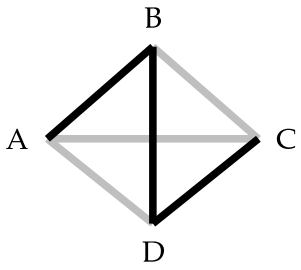
Vilain et al. (1995) introduce their approach with the following example.

- *Key:* $\langle A-B \quad B-C \quad B-D \rangle$
- *Response:* $\langle A-B \quad C-D \rangle$

The elements of the key yield an equivalence class that contains all four entities: $\{A, B, C, D\}$ (this is because all nodes are connected with B). The equivalence class can be represented by a fully connected graph with links between all elements, as shown below (black and gray links). The black links furthermore indicate a *minimal spanning tree*, i.e. a subgraph of the graph which is a smallest possible proper tree that contains all nodes:

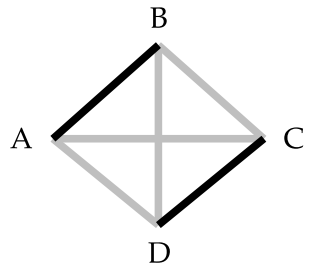


Minimal spanning trees are not unique, the one below would be a possible one for the equivalence class as well:



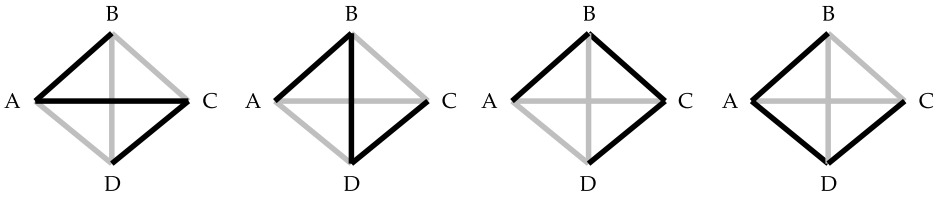
The number of links in all minimum spanning trees is the lowest possible and the same for all minimum spanning trees. In our example, this minimal number of links is 3.

In the response, the link between B and C is missing. The corresponding equivalence graph looks like this:



It is easy to see in this representation that the two links in the response are both correct – intuitively, one would expect a precision value of $\frac{2}{2} = 1$. The purely syntactic measure however arrives at the value of $\frac{1}{1+1} = \frac{1}{2}$.

In order to make the response match the key, one additional link must be added to the response. As seen before, there are multiple possible minimal spanning trees, in our example, there are four:



So, intuitively, in the response, two out of three links to be found are present, which amounts to a recall of $\frac{2}{3}$.

Given this graph representation, Vilain et al. (1995) introduce two new definitions for precision and recall. Recall is defined as the number of links in the key missing from the response, and precision is the number of links in the response missing from the key.

It is not feasible to calculate the missing links directly on the graph representation. An algorithm doing so would in the worst case have to consider all possible minimal spanning trees to find out the missing links. In more complex configurations, the complexity of this combinatorial problem might easily explode. Therefore, Vilain et al. (1995) suggest an alternative approach. They consider the *partitions* of the equivalence classes of the key and the response, which are unique.

Given an equivalence set K generated by the key, and a number of equivalence sets $R_1 \ldots R_n$ generated by the response, Vilain et al. (1995) define three functions:

- $p(K)$ is a partition of K relative to the response. Each subset in p is the result of intersecting K with those R_k that overlap K.

  If the key is $K = \{A, B, C, D\}$, and $R = \{A, B\}$, then $p(K) = \{\{A, B\}, \{C\}, \{D\}\}$.

- $c(K)$ is the minimal number of correct links to connect all elements in K. Obviously, this number is one less than the cardinality of K:

$$c(K) = |K| - 1$$

- $m(K)$ is the number of missing links in the response relative to the key. This is one less than the number of elements in the partition:

$$m(K) = |p(K)| - 1$$

The quantity $c(K) - m(K)$ is the number of links found in the response, which is the difference of the minimal number of correct links in the key $c(K)$ and the the number of missing links in the response $m(K)$.

Recall is then:

$$\begin{aligned} \mathit{Recall} &= \frac{c(K) - m(K)}{c(K)} \\ &= \frac{(|K| - 1) - (|p(K)| - 1)}{|K| - 1} \\ &= \frac{|K| - |p(K)|}{|K| - 1} \end{aligned}$$

Precision is the ratio of the number of correct links in the response and the total number of links in the response. The correct links is the subset of links in the response that also occur in the key. So, flipping views, precision is equal to recall with the response taken as the key and the key as the response. Therefore, we get the following "reverse" definition.

Given an equivalence set R for the response and a number of equivalence sets $K_1 \ldots K_n$ generated by the key, the "reverse" functions for calculating precision are defined as follows:

- $p(R)$ is a partition of R relative to the key. Each subset in p is the result of intersecting R with those K_k that overlap R.

- $c(R)$ is the minimal number of correct links to connect all elements in R:

$$c(R) = |R| - 1$$

- $m(R)$ is the number of missing links in the key relative to the response:

$$m(R) = |p(R)| - 1$$

Precision is then calculated as:

$$\begin{aligned} Precision &= \frac{c(R) - m(R)}{c(R)} \\ &= \frac{(|R| - 1) - (|p(R)| - 1)}{|R| - 1} \\ &= \frac{|R| - |p(R)|}{|R| - 1} \end{aligned}$$

The formulae for recall and precision apply to one key or one response, respectively. For calculating precision and recall over a whole corpus, it is sufficient just to sum over the individual results:

$$Precision = \frac{\sum_i (|R_i| - |p(R_i)|)}{\sum_i (|R_i| - 1)}$$

$$Recall = \frac{\sum_i (|K_i| - |p(K_i)|)}{\sum_i (|K_i| - 1)}$$

4.5 Functional evaluation

In the introduction to this chapter, we established that a prerequisite for the results produced by an evaluation strategy to be usable and reliable is their *relevance* with respect to the given task. Müller (2008) as well as Stuckardt (2001) argue that for research work that focuses on the task of anaphora resolution by itself, it may be relevant to evaluate the presence and correctness of any single link between two markables regardless whether they are pronouns, common or proper nouns. However, if the goal is to actually deploy the resolution system as a module in a larger application, the relevance of what to evaluate may shift and focus on whether the actual performance is good for achieving the desired function. Müller calls this evaluation strategy "Functional Evaluation".

Müller (2008) emphasizes that in his system for the resolution of *it*, *this*, and *that* in spoken dialog, what is actually important in order for a pronoun to be subsequently processible is that it is resolved to a *true non-pronominal expression*, i.e. an expression that actually bears meaning.

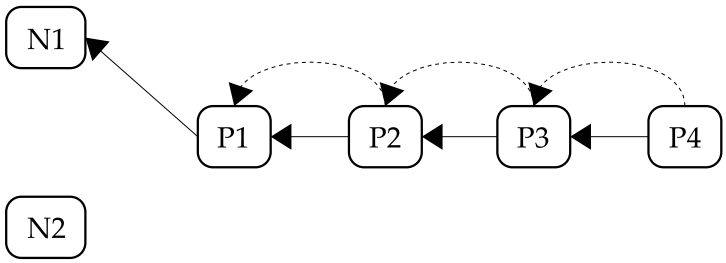


Figure 4.5: Chain of pronouns that are not linked to a non-pronominal expression. Solid lines: links in the key; dashed lines: links in the response (Müller, 2008).

Figure 4.5 shows a principled example. There are four pronouns P1-P4 which all belong to the same coreference chain. The head of the chain in the key, as indicated by the solid arrows, is N1, a non-pronominal expression. In the response, which is represented by the dashed arrows, this link between P1 and N1 is missing. The scoring scheme by Vilain et al. (1995) would yield a high recall of 75% since 3 out of the four required links were correctly found by the system. However, from the point of view of an actual application, the response generated by the system in this example is entirely useless, since actually *none* of the pronouns has been resolved, as the only elements in the coreference chain are all semantically empty pronouns. In an actual application, the most important information is which referent a pronoun actually refers to.

Müller (2008) points out that a similar effect occurs if there was a single wrong link from P1 to N2. Since all other pronouns depend on this link for their interpretation, *all* pronouns are resolved to the wrong antecedent, yielding a precision of 0 instead of a precision of 75% as would be reported by Vilain et al.'s measure. From the point of view of an application and the aspect of intuitiveness, the results reported by Vilain et al. would be very unintuitive.

For these reasons, Müller suggests to use slightly modified variants of precision and recall which implicitly apply a transitive closure to chains of pronouns, and thus only consider links between pronouns and non-pronominal antecedents:

$$Precision = \frac{\text{Correctly resolved pronouns}}{\text{All resolved pronouns}}$$

and

$$Recall = \frac{\text{Correctly resolved pronouns}}{\text{All resolvable pronouns}}$$

where

Correctly resolved pronouns is the number of pronouns in the *response* that are linked (directly or by transitive closure) to the *correct* non-pronominal antecedent,
All resolved pronouns is the number of pronouns in the *response* that are linked (directly or by transitive closure) to a non-pronominal antecedent, and
All resolvable pronouns is the number of pronouns in the *key* that are linked (directly or by transitive closure) to a non-pronominal antecedent.

4.6 Summary and conclusion

In this chapter, we discussed a number of approaches for evaluating pronoun resolution systems. The evaluation step is vital for any system, as only by proper evaluation it is possible to gain insight into the actual strengths and weaknesses of a system and compare its performance to that of other systems. In the introduction to this chapter, we established three basic properties that a useful evaluation strategy must have: *Applicability* concerns the preconditions on structural properties of the data that must be present in order for an evaluation approach to be usable. An evaluation strategy should produce *relevant* results, that means the performance figures it produces should be significant with respect to the goal which is to be reached. Finally, the results should also be *intuitive*, i.e. enable the scientist to interpret the results and get an adequate insight into the performance of the system.

Precision and recall are widely used as a scoring pattern, together, they characterize a system on the somewhat orthogonal dimensions of how accurate the system performs on those samples it handled, and how many samples the system is able to handle at all.

We distinguished between *link-based* and *class-based* scoring schemes for anaphora resolution. The only difference between the two is that link-based systems evaluate for a given pair whether the link connecting the two elements is correct, while class-based systems examine whether the two elements belong to the same class.

We then discussed the MUC-6 model theoretic scoring scheme by Vilain et al. (1995), which, unlike pair-wise evaluation strategies, looks at *full coreference sets* and gives a performance measure based on the minimum number of steps that would be necessary to align an erroneous response with the key.

Finally, we discussed the functional evaluation approach by Müller (2008), who observes that when a system for pronoun resolution is to be deployed in a practical system, the most important feature is that pronouns are resolved to an actually referring entity. A chain of pronouns with a missing non-pronominal head still remains an unresolved, semantically empty, chain.

4.6.1 Evaluation in this thesis

Anticipating the in-depth discussion of our data formats and resolution systems that will follow in chapters 6 and 8, we had the choice of evaluating the output of our systems either with a pair-wise approach or with the MUC-6 scheme. We did not employ the functional evaluation strategy, since our system is not to be deployed in a larger application at this point.

The core data structure that the MUC-6 scoring approach works on are *full coreferential chains*, i.e. coreferential chains that both contain pronouns linked to their antecedents with a relation of anaphora, but also non-pronominal (definite) NPs which are linked by a relation of coreference. The links in the key and the response may be arranged differently, and the advantage of the MUC-6 approach is that it deals with these different links efficiently and generates an intuitive report of performance.

Due to a preset requirement on the research domain of this thesis, neither the rule-based system in chapter 6, nor the hybrid system in chapter 8 generate full coreferential chains. Instead, they exclusively generate links of anaphora or cataphora between a pronoun and an antecedent. A pronoun may actually be linked to more than one antecedent, in which case multiple pairs involving the same pronoun but different antecedents occur

in the output, but they are always treated as separate pairs. Both systems do not create links of coreference between definite NPs. Thus, in the absence of full coreferential chains, the MUC-6 approach does not meet the applicability requirement described in the introduction to this chapter – although using it would not do any harm, as with a purely pairwise data structure, the computations of the MUC-6 approach are reduced to computations equivalent to plain pair-wise approaches.

In this light, the preferred evaluation strategy is a pair-wise one, both because its application is more intuitive given the data, and the implementation complexity of the MUC-6 approach would not be justifiable given its limited applicability. We therefore chose to use the class-based pair-wise evaluation strategy as described in section 4.2.2.

Chapter 5

The Data

5.1 The TüBa-D/Z treebank

The Tübingen Treebank of Written German (*Tübinger Baumbank des Deutschen/Zeitungstext*, abbreviated TüBa-D/Z; Telljohann et al. 2006) is the central source of linguistic data that is used by all experiments that will be carried out in this thesis. The raw text is taken from the German daily newspaper *die tageszeitung*. The third release of the treebank, which is the version considered in this thesis, comprises 27125 sentences (473747 tokens).

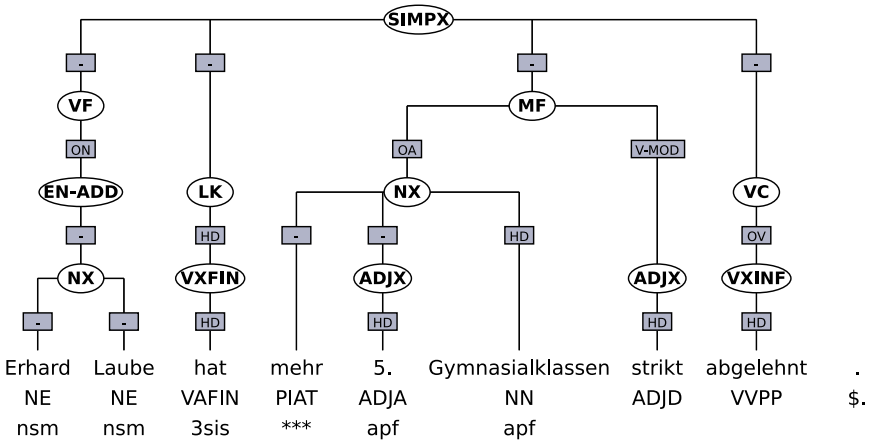


Figure 5.1: A sample tree from the TüBa-D/Z treebank.

Figure 5.1 shows an annotated example sentence from TüBa-D/Z. In the annotation of this sentence, several layers of annotation can be distin-

guished:

- **the word level**
- **the level of phrases**
- **the level of topological fields**
- **the clause level**

In what follows, we will concentrate on the parts of the annotation that are relevant to anaphora resolution. A comprehensive description of the treebank's annotation scheme can be found in the TüBa-D/Z stylebook (Telljohann et al., 2006).

5.1.1 The word level

Linguistic annotation on the **word level** comprises three parts, as illustrated in figure 5.1. Firstly, the tokens of the text proper are located on the word level. All other annotation in the treebank is to be interpreted relative to the sequence of tokens. Secondly, tokens are annotated with information about their parts of speech (on the POS level) and, thirdly, their inflectional properties (on the morphological level). For the annotation of parts of speech, TüBa-D/Z utilizes the Stuttgart-Tübingen POS-tagset (STTS; Schiller et al. 1999). The multi-letter POS tags in STTS are hierarchically structured with the main part of speech usually being encoded in the first letter, such as **N** for noun, or **P** for pronoun. The code for the main part of speech is followed by more specific information, such as **NN** for common noun, **NE** for proper noun *(Eigenname)*, or **PRF** for reflexive pronoun. The subset of part of speech tags that is most relevant for the task of pronoun resolution is listed in table 5.1.

POS	Description	Examples
NN	common noun	*Tisch, Herr, [das] Reisen*
NE	proper noun	*Hans, Hamburg, HSV*
PPER	personal pronoun	*ich, er, ihm, mich, dir*
PPOSAT	attributive possessive pronoun	*mein [Buch], deine [Mutter]*
PRF	reflexive pronoun	*sich, einander, dich, mir*

Table 5.1: Subset of STTS most relevant for pronoun resolution. See appendix A for a complete description of the STTS tagset.

The tokens in the example sentence in figure 5.1 have the POS tags

*Erhard/**NE** Laube/**NE** hat/**VAFIN** mehr/**PIAT** 5./**ADJA** Gymnasialklassen/**NN** strikt/**ADJD** abgelehnt/**VVPP** ./**$.***

This indicates a sequence of two proper nouns *'Erhard'* and *'Laube'*, a finite auxiliary *'hat'*, an indefinite pronoun *'mehr'*, an adjective *'5.'*,[1] a noun *'Gymnasialklassen'*, an adverb *'strikt'*, a participle *'abgelehnt'*, and finally the period (punctuation characters count as separate tokens).

In addition to the POS analysis, each token is assigned a label that encodes its inflectional (morphological) status. For nouns, this is a three letter code where the first letter represents the noun's case, the second letter its number and the third letter its gender, as illustrated in figure 5.2.

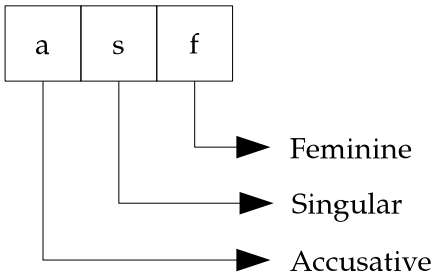


Figure 5.2: Sample inflectional tag for nouns

In the example in figure 5.1, both parts of the named entity *Erhard Laube* have the morphological labels `nsm`, which stands for *nominative singular masculine*.

For pronouns, a fourth slot is added to the label, indicating the pronoun's person (figure 5.3).

If a morphological feature is underspecified, a star (*) may be inserted into a slot. Table 5.2 lists all applicable[2] values.

A complete chart of all valid combinations of morphological tags and POS tags can be found in appendix B.

[1] *'5.'* is here an abbreviated notation for *'fünfte'*. This is why the punctuation character remains attached to the digit, which usually would be a violation of the segmentation rules in TüBa-D/Z.

[2] Values for verbal material, such as tense and mood information are omitted here.

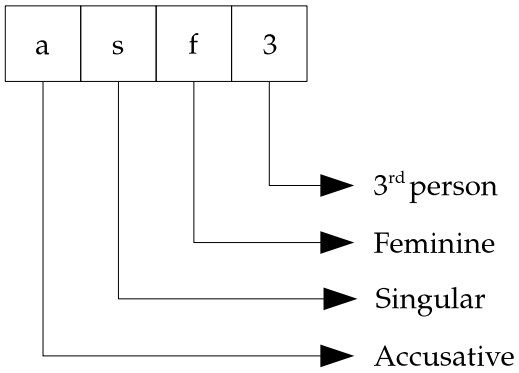


Figure 5.3: Sample inflectional tag for pronouns

Feature	Values
case	n (nominative), g (genitive), d (dative), a (accusative), * (underspecified)
gender	m (masculine), f (feminine), n (neuter), * (underspecified)
number	s (singular), p (plural), * (underspecified)
person	1 (first), 2 (second), 3 (third), * (underspecified)

Table 5.2: Values of morphological features

5.1.2 The level of phrases

Phrasal syntactic annotations, such as noun and verb phrases are located on the phrase level. Phrasal categories may be nested; however their annotation adheres to the *flat clustering principle* which is one of the central principles of annotation in TüBa-D/Z. The flat clustering principle favors flat annotation structures over deeply nested structures. This results in arbitrary branching. The *longest match principle* requires that as many daughter nodes as possible are combined into a single mother node, provided that the resulting construction is both syntactically and semantically well formed. The *high attachment principle* finally demands that syntactically and semantically ambiguous modifiers are always attached to the highest possible level in a tree structure, avoiding for example the problematic decision whether a postmodifier is a free adjunct or a complement of the modified phrase.

In figure 5.1, the nodes that belong to the level of phrases can be iden-

tified by the letter X that they contain in the category label (except SIMPX, which belongs to the clause level). Appendix C lists all possible category labels in TüBa-D/Z.

5.1.3 The structure of noun phrases

The most important kind of phrase for anaphora resolution is the noun phrase. This section will summarize the principles of annotation of noun phrases in TüBa-D/Z. Chapter 4 of the TüBa-D/Z stylebook (Telljohann et al., 2006) discusses the annotation scheme for NPs in greater detail.

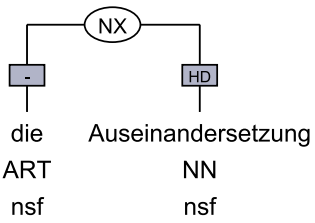
The annotation scheme distinguishes two basic types of noun phrases: simple noun phrases, and complex noun phrases.

Simple noun phrases consist of a head noun which may be a common noun, a proper noun, or a pronoun. The head noun may optionally be preceded by a determiner and adjectival or nominal premodifiers of arbitrary complexity.

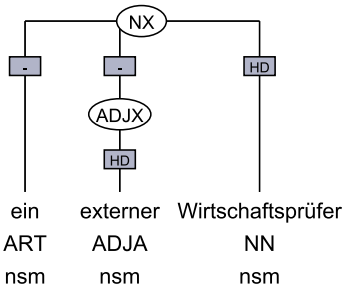
Complex noun phrases are simple noun phrases with one or more postmodifiers of any syntactic category and arbitrary complexity.

Prenominal modification

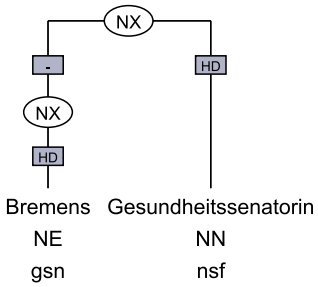
Following the flat clustering principle, all premodifying material is attached to the NX on the same level as the head, as below.



The head noun *Auseinandersetzung* is assigned the function label head HD. The determiner is assigned the empty label -. Prenominal modifiers can either be attributive adjectives:



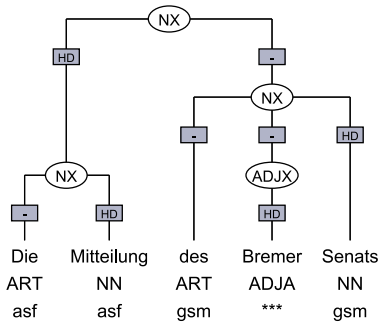
or preceding genitive phrases:



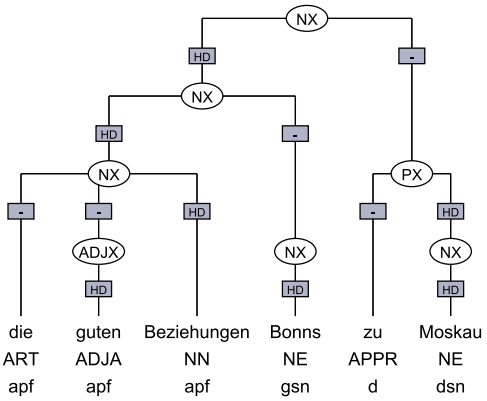
In figure 5.1, the noun phrase *mehr 5. Gymnasialklassen* is premodified by the adjective phrase *5.* and the indefinite pronoun *mehr*.

Postnominal modification

Unlike premodifiers, which are attached to the noun phrase on the same level as the head, postmodifiers are always projected to the phrase level and then attached on the next higher level. In the following example, the genitive phrase *des Bremer Senats* postmodifies the head *die Mitteilung*. The modifier phrase is attached on the next higher level NX:



If a noun has multiple postmodifiers, they are arranged in a hierarchical structure with each modifier modifying the preceding NX:

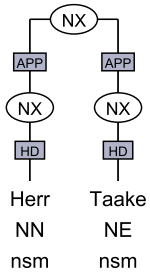


Appositions

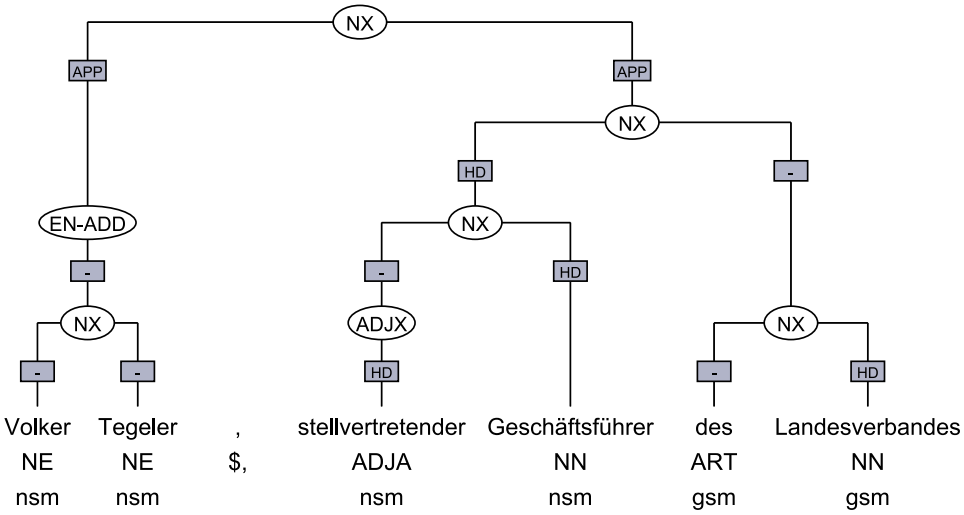
Bußmann (2002) defines the term *apposition* as follows:

> An apposition is a facultative constituent of a noun phrase, which agrees syntactically and usually referentially with the nominal kernel.

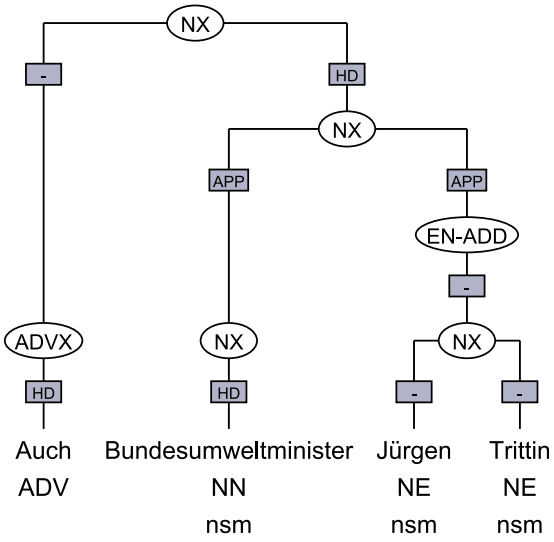
As stated in the TüBa-D/Z stylebook (Telljohann et al., 2006), there is no general agreement in the literature about the exact definition of *apposition*. Specifically, grammarians do not concur in which part is the head noun and which part is the apposition. Therefore, the TüBa-D/Z annotation scheme avoids the distinction between head and apposition in appositional constructions. All parts are first projected to the phrase level, and then coordinated, receiving the APP label. The mandatory criterion for an appositional structure is that all parts are referentially identical, as in the following example.



The following sentence is an example of a more complex apposition of a proper noun and a complex NX:



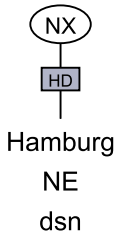
Deviating from the general annotation scheme, premodifiers of the whole appositional construction are attached to the next higher level NX:



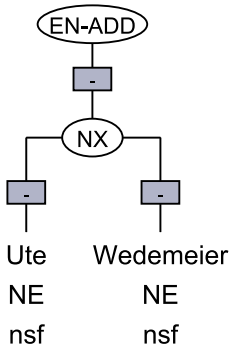
Proper nouns and named entities

Proper nouns denote individual living beings, objects and so on, which are unique entities with their own specific properties (Telljohann et al., 2006).

The TüBa-D/Z annotation scheme defines three ways of annotating proper nouns. A proper noun that consists of a single word, such as *Hamburg*, is tagged `NE` on the POS level, and projected to an NX on the phrase level:

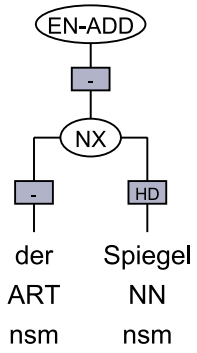


More complex syntactic constructions that denote proper nouns are called *named entities* in the TüBa-D/Z annotation scheme. On the phrase level, they are first projected to an NX node, and then an additional EN-ADD category node is inserted to mark the constituent as a named entity:



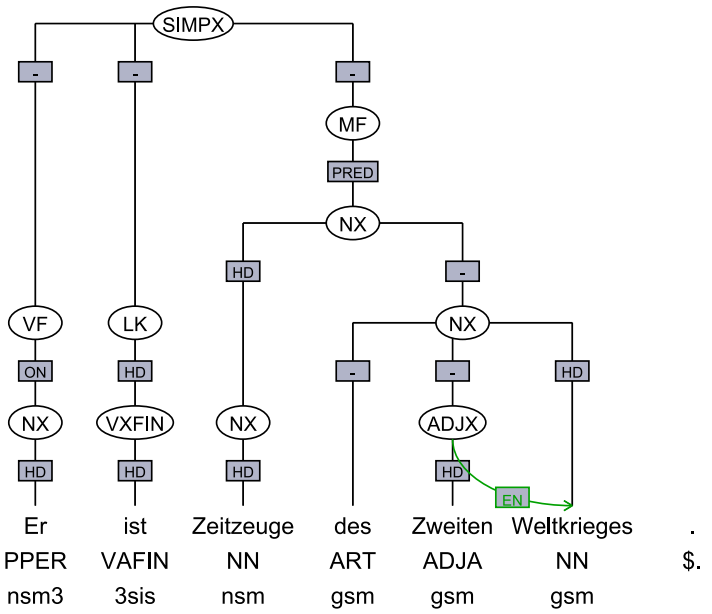
The subject in figure 5.1 has the same syntactic structure.

If a complex syntactic structure, such as a phrase or a sentence denotes a named entity, the whole phrase/sentence is annotated as usual and then projected to an EN-ADD node:

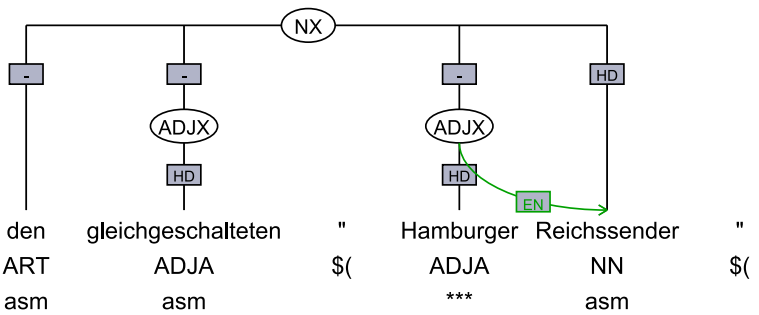


No EN-ADD node is inserted into the syntactic tree if the original form of the proper noun is inflected or premodified, such as *Zweiter Weltkrieg*, which appears as *des Zweiten Weltkriegs* in the following example. In such

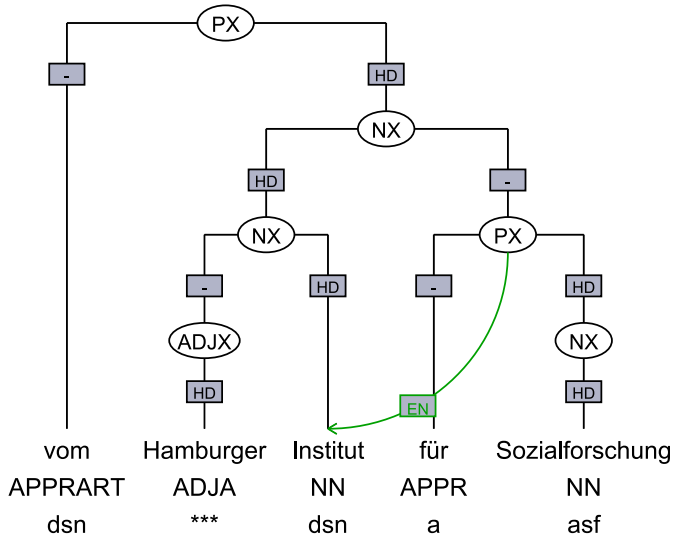
cases, a secondary edge labeled EN is annotated that points from the dependent part of the proper noun to its head noun:



In the following example, the proper noun *Hamburger Reichssender* is premodified by a adjective and a determiner. Because of the flat clustering principle, there is no constituent that spans *only* the proper nouns *Hamburger Reichssender*. Therefore, an EN-ADD node cannot be inserted here, and a secondary edge is used to mark up the named entity.



A postmodifier that is part of a proper noun is marked by the EN secondary edge as well:



5.1.4 Syntactic annotation of pronouns

The STTS POS tagset (Schiller et al., 1999) distinguishes between two main types of pronouns: *substituting* pronouns that occur *in place of an NP*, and *attributive* pronouns that occur *within an NP*. Substituting pronouns replace a full NP. Therefore, they are projected to their own NX node on the phrase level (see figure 5.4). Attributive pronouns usually occur as premodifiers and are attached to the same NX as the head they modify (see figure 5.5).

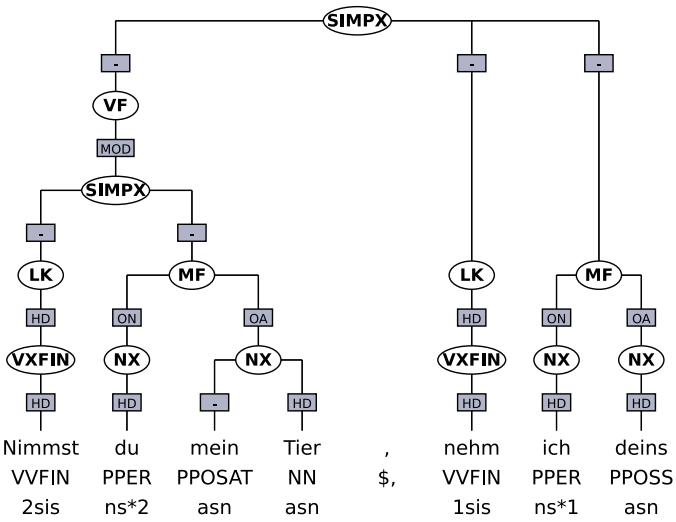


Figure 5.4: Substituting possessive pronoun *deins/PPOSS*

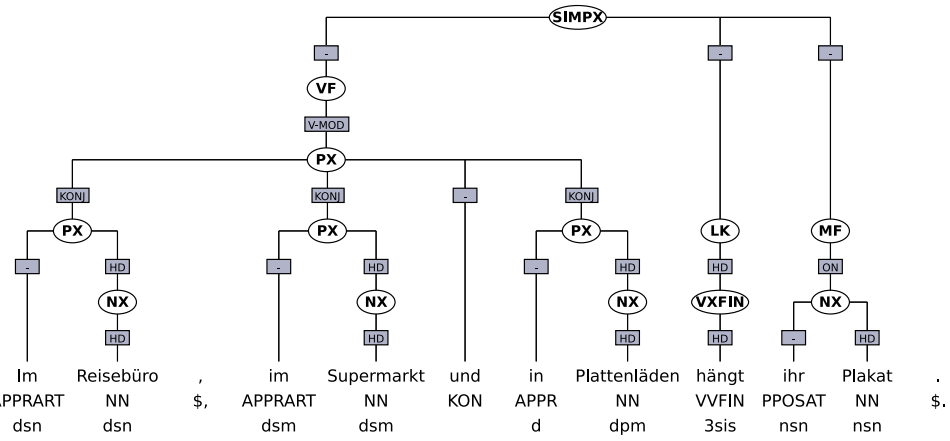


Figure 5.5: Attributive possessive pronoun and premodifier *ihr/PPOSAT*

Personal pronouns and reflexive pronouns are always substituting, therefore they are projected to an NX node.

The pronoun *es*

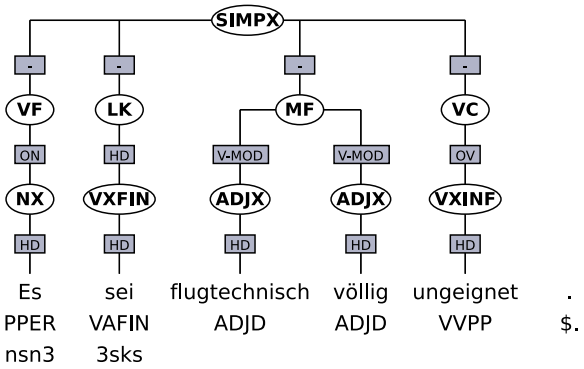
The pronoun *es* – analogously to the English *it* – can either function as a neuter personal pronoun or as an expletive pronoun. Its distribution is slightly more complex than in English and of special importance to pronoun resolution, as instances of expletive *es* need – and should – not be resolved to any antecedent. The following cases are distinguished in the TüBa-D/Z annotation scheme.

Personal pronoun The pronoun is in referential identity with some real-world entity, as in the sentence

[Es] sei flugtechnisch völlig ungeeignet

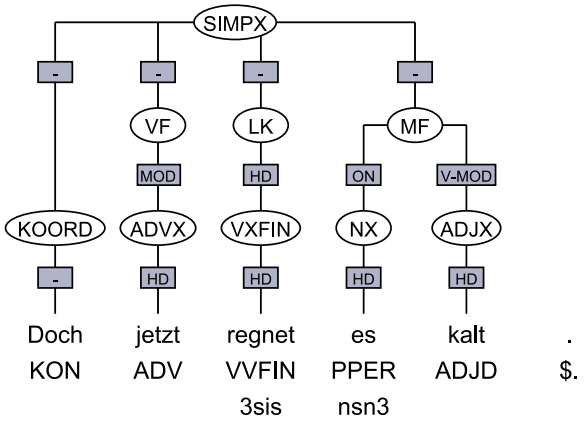
where the pronoun refers back to the neuter NP *das geräumte Gelände* in the preceding sentence

Busch erklärte gleichzeitig, es sei "völlig ausgeschlossen", daß [das geräumte Gelände] im Süden des Flughafens für den Bau der umstrittenen neuen Landebahn genutzt werden könne.



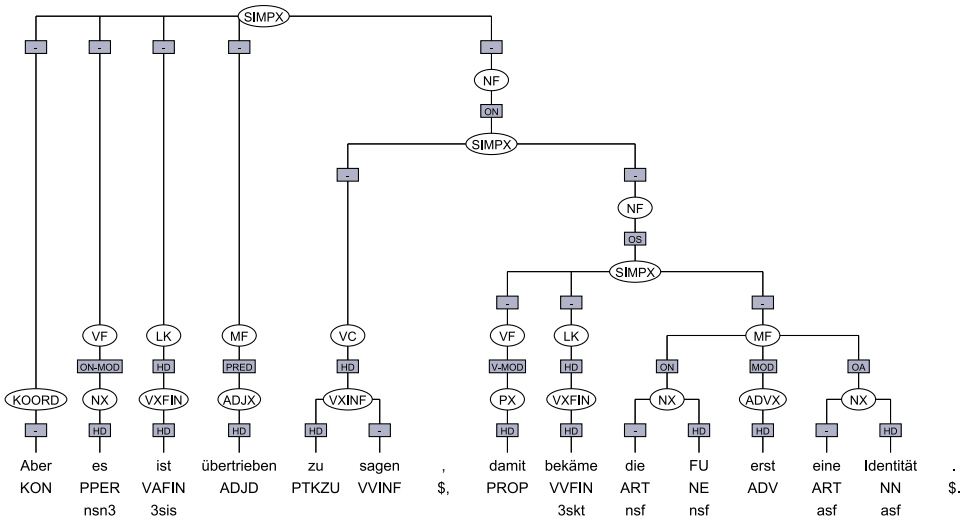
Formal subject or object The pronoun *es* occurs as a formal subject or object with verbs that syntactically require the argument slot of the subject or object to be filled, while not assigning any semantic function to it. An example for verbs that obligatorily subcategorize for an expletive subject are weather verbs and impersonal and agentless constructions such as *Es gibt so eine Buchung*.

The formal subject may be moved into the middle field, as the following example shows. The formal subject of the weather verb *regnet* is realized in the middle field:



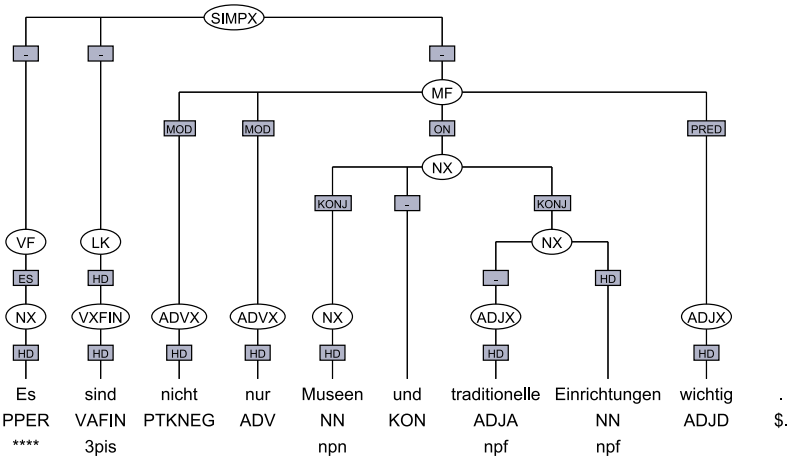
With respect to their grammatical function, formal subjects or objects cannot be distinguished from referential personal pronouns, both types are labeled `ON` or `OA`, respectively.

Correlate *es* A clausal argument that is extraposed in the final field may optionally be doubled by an expletive in the initial or middle field, which is labeled `ON-MOD` or `OS-MOD`, depending on the grammatical function of the extraposed clausal argument:



The pronoun *es* is located in the initial field, and bears the function label `ON-MOD`. The actual subject of the sentence, the embedded sentence *zu sagen, damit bekäme die FU erst eine Identität*, was moved into the final field.

Vorfeld-*es* This type of expletive *es* only occurs in the initial field and is a purely structural dummy element which is not correlated with any argument of the clause. Its distribution is restricted to the initial field position. The TüBa-D/Z annotation scheme stipulates the grammatical function label `ES` for *Vorfeld-es*.



Eisenberg (1999, p. 175–176) provides an extensive discussion of the distinction between Correlate *es* and Vorfeld-*es*.

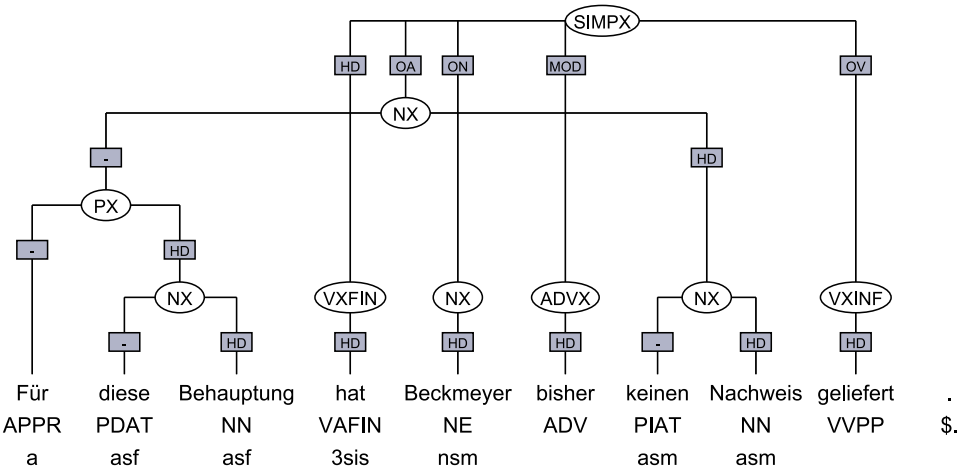


Figure 5.6: A tree with hypothetical crossing branches

5.1.5 The higher syntactic levels

Above the phrase level, the TüBa-D/Z contains three additional syntactic levels: **the level of topological fields**, the level of grammatical functions, and finally the level of clauses. The inclusion of a level of topological fields is a consequence of the relatively free word order in German. The theory of topological fields (Höhle, 1986) is a purely descriptive theory that partitions a sentence into multiple slots, called *topological fields*. Two slots, the *left* and *right sentence bracket* (also called *verb complex*), may be filled with verbal material. They constitute the structuring foundation of the sentence. The field before the left sentence bracket *(Vorfeld/initial field)*, contains exactly one phrase. The field between the sentence brackets *(Mittelfeld/middle field)*, and after the right sentence bracket *(Nachfeld/final field)* may contain an arbitrary number of phrases in virtually any order. Embedded clauses can be placed in these fields as well.

In the example sentence in figure 5.1 at the beginning of this chapter, four field positions are occupied: The subject *Erhard Laube* is located in the initial field (with category label `VF`, *Vorfeld*), and the finite (auxiliary) verb is positioned in the left sentence bracket (`LK`, *Linke Klammer*). The middle field (`MF`, *Mittelfeld*) contains the accusative object *mehr 5. Gymnasialklassen* and the adverb *strikt*, which modifies the verb. The past participle *abgelehnt* finally is in the verb complex `VC`.

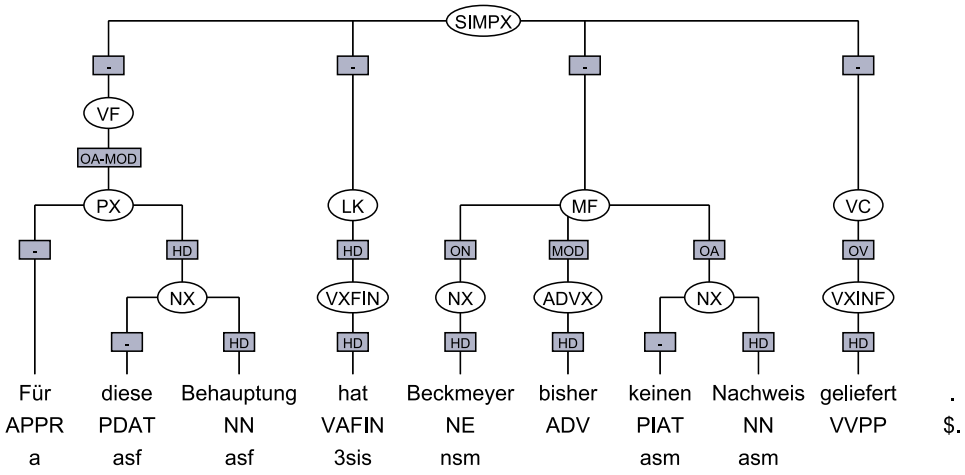


Figure 5.7: A sample tree from TüBa-D/Z without crossing branches.

For the TüBa-D/Z annotation scheme, it has been decided that no crossing branches are allowed in syntactic trees. Due to the free word order in German, it is offhand possible for any syntactic entity, such as as a verb complement, to be displaced in a position other than the one expected by default order. Given no other means of representation, crossing branches are inevitable in such cases. A hypothetical example for this configuration is shown in figure 5.6. The prepositional phrase *für diese Behauptung*, which is a prepositional complement to the NP *keinen Nachweis* has been fronted, which would result in a crossing branch between the PX and the NX which remained in place.

In TüBa-D/Z, the dependency structure is expressed by the edge labels instead. If a phrase is extraposed, the edge label clearly states its grammatical function even though no explicit connecting branch exists between this phrase and the syntactic element that this phrase modifies or that it is a complement of. This is shown in figure 5.7. Unlike the tree in figure 5.6, there are no crossing branches in this tree. The PX is annotated as being located in the *Vorfeld* of the sentence, and its grammatical function label is OA-MOD, which indicates that it is a modifier of the accusative object *keinen Nachweis*. See appendix D for a list of all possible edge labels.

5.2 Annotation of coreference in TüBa-D/Z

5.2.1 Markables

Poesio (2004) defines *markables* to be "the text constituents that realize semantic objects that may enter in anaphoric relations". In other words, a markable is the fraction of a text that represents an extra-linguistic entity (or referent) in the text. When annotated in a corpus, it is the markables that are the start and end points of referential relations that hold between their corresponding referents. The exact definition of what counts as a markable is left up to the concrete annotation model. In the TüBa-D/Z treebank, the term markable is determined entirely by the syntactic annotation that is available on the phrasal level. This is advantageous in a number of respects:

- By relying on the present syntactic annotation, it is possible to *automatically* select the relevant fractions of the text for markables. By means of this, it is possible to skip one laborious step of manual annotation.
- The automatic suggestion of markables eliminates one source of errors which are introduced by annotators selecting spans of text that do not adhere to the annotation guidelines.
- Furthermore, the definition of the term markable on the grounds of existing syntactic annotation reduces the number of different linguistic entities in the treebank and ensures that the individual levels of annotation are compatible to each other.

In TüBa-D/Z, only nominal elements are considered markables. Other syntactic constituents, such as verbal material, are ignored. The following rules are applied for extracting markables:

- All noun phrases are markables with category NX. The resulting markables span all tokens that are dominated by the NX node. Nested noun phrases yield multiple markables with different spans. From the example sentence in figure 5.8, five markables would be extracted:

  1. *Volker Tegeler , stellvertretender Geschäftsführer des Landesverbandes*

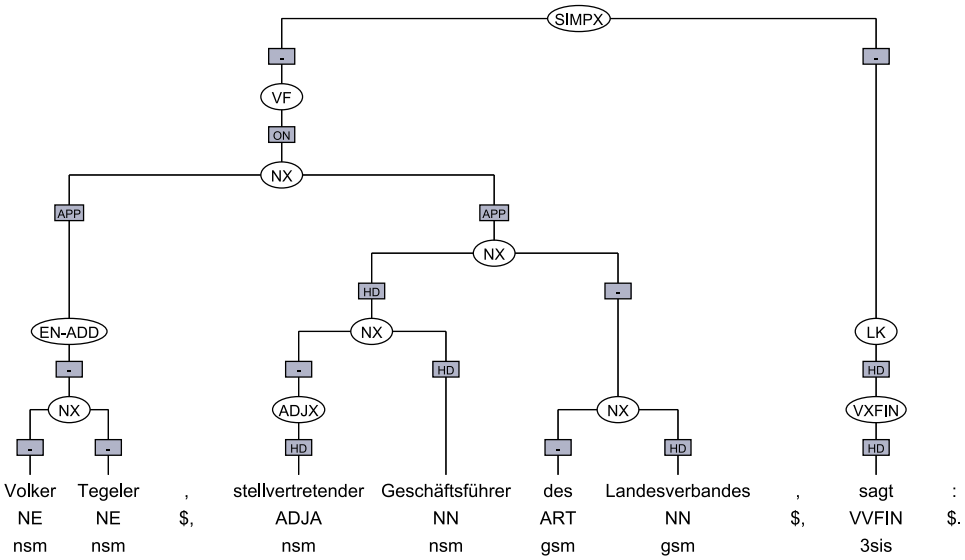


Figure 5.8: Nested noun phrases that yield multiple markables

2. *Volker Tegeler*
3. *stellvertretender Geschäftsführer des Landesverbandes*
4. *stellvertretender Geschäftsführer*
5. *des Landesverbandes*

- All pronouns are markables. As discussed in section 5.1.4, there are two kind of pronouns: substituting and attributive. For substituting pronouns there is a unary projection to an NX, therefore the first rule automatically applies. Attributive possessive pronouns (STTS tag PPOSAT), such as *sein/PPOSAT Auto* and attributive relative pronouns (STTS tag PRELAT) are not projected to unary NX parent nodes, but markables are extracted for these pronouns as well.
- No other syntactic constituents are markables.

Two final remarks are to be made about markables. Firstly, as pointed out by Hirschmann and Chinchor (1997), the *presence* of a markable does not necessarily imply that it is also part of a referential relation. Secondly, the TüBa-D/Z annotation guidelines include a *longest match rule* that requires that if there exist nested markables with the same head, only the *longest* markable is annotated to be the source or target of a referential relation.

5.2.2 Referential relations

Van Deemter and Kibble (2001) distinguish two basic types of referential relations: Two referents α_1 and α_2 are *coreferent* if and only if they refer to the same extralinguistic referent. They also give a definition of anaphoricity: The two referents are *anaphoric* if and only if α_1 depends on α_2 for its interpretation. The annotation model for referential relations on which the annotation of referential relations in TüBa-D/Z is based on Naumann (2006) and includes relations of both types. As mentioned in the previous section on markables, in TüBa-D/Z only referential relations between nominal syntactic elements are annotated, but not between other syntactic types, such as verbal material. Therefore, anaphoric phenomena like *event anaphora* are excluded from the annotation.

The annotation scheme in TüBa-D/Z is loosely based on the inventory of categories that is suggested by the MATE meta scheme for the annotation of coreference (Poesio, 2000).

The annotation scheme of TüBa-D/Z distinguishes eight different types of referential relations. Most important are the two relations of coreference and anaphora. Six additional referential relations are annotated as specified below. The eight types of referential relations that are annotated in the treebank are:

- coreferential
- anaphoric
- cataphoric
- bound
- split antecedent
- instance
- expletive

We will now discuss each of these relations in greater detail.

Coreferential

The *coreferential* relation holds between two non-pronominal markables m_1 and m_2 if and only if both m_1 and m_2 refer to the same extra-linguistic referent. This definition is equivalent to the one by van Deemter and Kibble (2001) that was discussed in the previous section.

An example for a relation of coreference is given in (1). Here, the two NPs *Metropol* and *Theater* refer to the same extra-linguistic referent, a theater building.

(1) Der Vorhang geht wieder auf im [$_1$ **Metropol**]. Kultursenator will [$_2$ **das Theater**] an Privatinvestor verkaufen.

Anaphoric

The *anaphoric* relation holds between a non-pronominal or pronominal antecedent *a* and a pronoun *p*, as shown in example (2).

(2) [$_1$ **Ein klarer Ton**] breitet [$_2$ **sich**] aus, warm und satt, bis [$_3$ **er**] den ganzen Saal erfüllt. Dann dünnt [$_4$ **er**] aus, zerbröselt und verflüchtigt [$_5$ **sich**].

The marked NP and the pronouns in sentence (2) belong to the same coreference chain.[3] They all refer to the same referent *[Ein klarer Ton$_1$]*, which is the first element in the coreference chain. The reflexive pronoun *sich$_2$* is anaphoric to *ein klarer Ton$_1$*. The personal pronoun *er$_3$* in turn is anaphoric to *sich$_2$*, and so forth.

In German, some reflexive pronouns have a special status. They do not refer to any referent, but are solely a syntactic requirement of the verb such as in

(3) Ein schweres Erdbeben hat **[sich]** gestern ereignet.

This subtype of reflexive pronouns is called *inherently reflexive* and is not annotated. Pronouns are always assigned the label *anaphoric* even when they are coreferent to their antecedents.

[3] See chapter 3 for a discussion of the differences and similarities of the terms *coreference set* and *coreference chain*.

Cataphoric

A pronoun is in a *cataphoric* relation to its antecedent, if the antecedent *follows* the pronoun, i.e. the pronoun refers to an entity that is introduced into the discourse only *after* the pronoun (a more accurate term in this context would be *postcedent* instead of antecedent).

The main difference between cataphoric relations and anaphoric relations is therefore the position of the noun phrase that the pronoun refers to:

(4) 222 Tage währte **[**$_1$ **ihre]** kleine Traktoren-Mahnwache am Alexanderplatz. Mit drei landwirtschaftlichen Nutzfahrzeugen und ständiger personeller Präsenz im nebenstehenden Campingmobil dauerprotestierten **[**$_2$ **die Beschäftigten der Landtechnik Schönebeck (LTS) sowie MitarbeiterInnen des Tochterunternehmens GS Fahrzeug- und Systemtechnik]** für die Übernahme ihres Betriebes durch einen West-Investor.

Here, the pronoun *ihre* is cataphoric to the NP *die Beschäftigten der Landtechnik Schönebeck (LTS) sowie MitarbeiterInnen des Tochterunternehmens GS Fahrzeug- und Systemtechnik*, which occurs after the pronoun.

If an antecedent occurs in the headline of a text as the only antecedent prior to a pronoun, and another NP follows the pronoun, the a *cataphoric* relation is annotated between the pronoun and the second NP. Thus, cataphoric relations local to the text body overrule discourse-global relations.[4]

Bound

The *bound* relation holds between a pronoun p and an antecedent a if a is bound by a quantifier (Poesio, 2000), such as in the following examples:

(5) **[**$_1$ **Die meisten Benutzer]** kaufen **[**$_2$ **sich]** **[**$_3$ **ihre]** Tassen selbst.

In (5), the antecedent *[die meisten Benutzer*$_1$*]* contains a quantifier *die meisten*. The reflexive *sich*$_2$ is in a *bound* relation to the antecedent *[die meisten Benutzer*$_1$*]*. The possessive *ihren*$_3$ is *anaphoric* to *sich*.

[4] If the NP in the headline is the only antecedent in the whole article, then an anaphoric relation between the pronoun and the antecedent in the headline will be annotated.

Split antecedent

The *split antecedent* relation is annotated when a plural pronoun refers to two referents at once which are not adjacently realized in the text, as in example (6).

(6) "Vor allem die letzten Stunden waren fürchterlich", sagt **[$_1$ eine junge Frau]**, die **[$_2$ ihre gebrechliche Mutter und vier Kinder]** über die Grenze führt. **[$_3$ Sie]** sind zu Fuß gekommen, denn das Auto wure **[$_4$ ihnen]** von serbischen Freischärlern abgenommen.

The pronoun *sie*$_3$ refers to the union of the referents *eine junge Frau*$_1$ and *ihre gebrechliche Mutter und vier Kinder*$_2$. It is not possible to annotate a relation of *anaphoric* here, since there is no single NX that could serve as a markable in which the end point of the relation could be anchored.

Instance

If a pronoun or an NP refers to particular instance of a class of entities denoted by the antecedent, then an *instance* relation is annotated. In example (7), the pronoun *jene*$_2$ refers refers to a restricted subset *viele Banken*$_1$, those which are owned by the state:

(7) Doch **[$_1$ viele Banken]**, vor allem **[$_2$ jene]** im Staatsbesitz, sitzen auf Haufen von faulen Krediten.

Expletive

The category *expletive* is reserved for annotating the pronoun *es* if it is semantically empty, i.e. expletive, or pleonastic. The distribution of expletive *es* in German is similar to that of English expletive *it*, such as in weather verbs like *regnen* in *es regnet*.

Some instances of expletive *es* are annotated on the syntactic layer of TüBa-D/Z. These are the cases of *Vorfeld-es*, where the subject is realized in the middle field, and the otherwise unoccupied position in the initial field is filled with a semantically empty expletive *es*, as illustrated in figure 5.9. All occurrences of expletive *es* are automatically copied from the syntactic layer into the coreference layer.

Section 5.1.4 discusses the distribution of expletive *es* in greater detail.

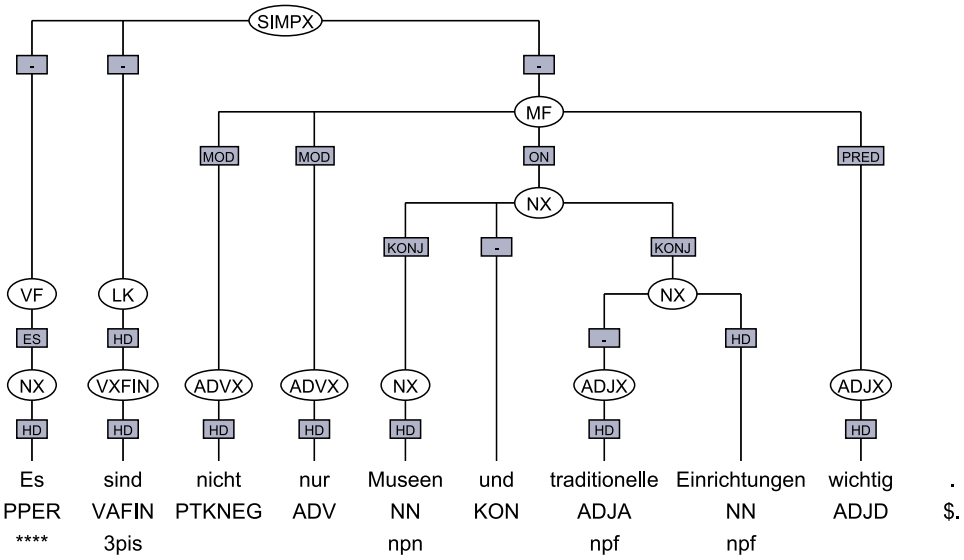


Figure 5.9: A sentence containing a *Vorfeld-es*.

5.3 A quantitative view of TüBa-D/Z

This section will discuss the quantitative properties of the TüBa-D/Z treebank and its annotation. All experiments and figures that are given in this thesis refer to the third release of the treebank.[5] TüBa-D/Z contains 27 125 sentences and a total of 473 747 tokens. The treebank contains 44 424 pronouns, which amounts to roughly 10% of the total number of tokens. The distribution of pronouns according to their parts of speech is shown in figure 5.10. The subset of pronoun types that is later considered for resolution comprises personal pronouns (PPER), attributive possessive pronouns (PPOSAT), and reflexive pronouns (PRF). These three types together make up 53% of all pronouns.

The syntactic units of noun phrases and pronouns together yield a total of 172 977 markables.

Figure 5.11 shows the frequency distribution of referential relations in the corpus. The two "major" types of referential relations, *coreferential* and *anaphoric* together comprise more than 87% of all relations in the corpus, with their relative order of magnitude being roughly equal. The next category, *expletive*, occurs less than a seventh of the number of coreferential

[5] Release 3 of TüBa-D/Z was published on 14/07/2006. We do not consider the newer release 4 of the treebank.

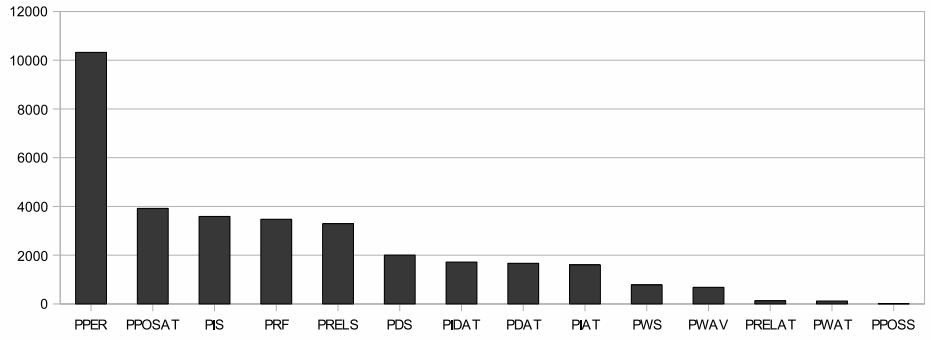


Figure 5.10: Distribution of the frequency of parts of speech of the pronouns in TüBa-D/Z (see appendix A for a full explanation of the POS labels)

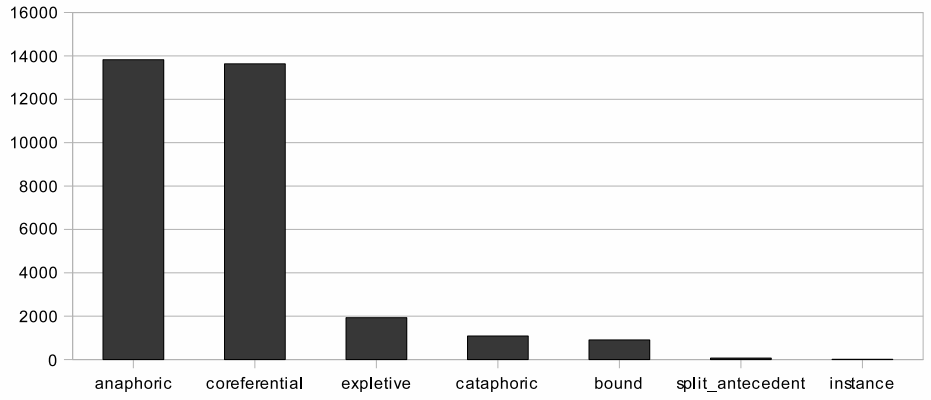


Figure 5.11: Distribution of referential relations in TüBa-D/Z

or anaphoric relations, and the remaining categories have even lower frequencies. This distribution is not surprising. Pronominalization is probably the most important means of expressing reference to an entity that has previously been introduced into the discourse. Equally important, and a specific stylistic property of newspaper text, is the referral to one and the same entity by multiple different nominal expressions - thereby conveying additional information or avoiding unaesthetic repetitions of the same term. Cataphoric relations are rather infrequent which reflects the fact that referring to an entity that has not yet been introduced into the discourse requires increased processing overhead which is – if it is valid to speak of an "economy of communication" in this context – suboptimal in this respect. As a consequence, cataphora is more of a means to produce specifically

Relation	Frequency
anaphoric	13 824
coreferential	13 635
expletive	1 930
cataphoric	1 085
bound	906
split_antecedent	67
instance	11

Table 5.3: Distribution of referential relations in TüBa-D/Z

marked stylistic utterances.

Table 5.3 lists the numbers of occurrence of referential relations in TüBa-D/Z.

5.4 The TüPP-D/Z treebank

The *Tübingen Partially Parsed Corpus* TüPP-D/Z (Müller, 2004b) has been automatically annotated using the cascaded finite state parser KaRoPars (Müller, 2007). Just like its sister treebank TüBa-D/Z, TüPP-D/Z is based on the German daily newspaper *die tageszeitung*. Due to its automatic annotation, TüPP-D/Z is several orders of magnitude larger than TüBa-D/Z and contains more than 11.5 million sentences (about 200 million tokens).[6] On the other hand, no manual error correction was performed during the annotation process. The corpus contains four levels of syntactic constituency: the lexical level, the chunk level (in this respect, TüPP-D/Z differs from TüBa-D/Z), the level of topological fields, and the clausal level. Unlike TüBa-D/Z, which assumes a relatively deep syntactic structure, trees are quite flat in TüPP-D/Z. Due to limitations of the finite state parsing model, the attachment of chunks remains underspecified. Major constituents are annotated with grammatical functions. Figure 5.12 shows an example sentence from TüPP-D/Z. The categories indicating the left and

[6]The variant of TüPP-D/Z without grammatical functions contains 204 661 513 tokens. Due to technical difficulties in the annotation component for grammatical function beyond the author's control, the GF version of TüPP-D/Z is slightly smaller than the non-GF version. It comprises 194 826 942 tokens.

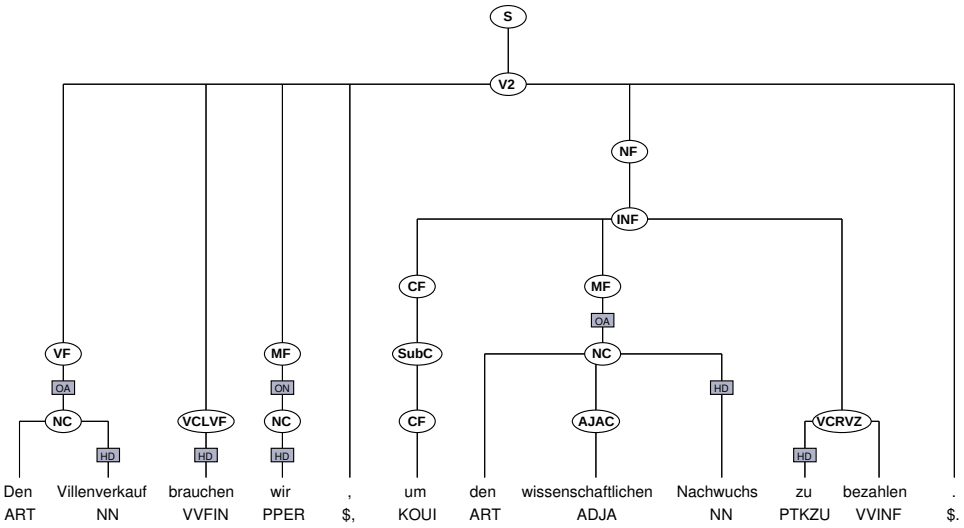


Figure 5.12: A sample from the automatically annotated TüPP-D/Z treebank.

right sentence brackets are merged with the categories of verb chunks.

Although the annotation of TüPP-D/Z provides less syntactic structure than TüBa-D/Z, the relevant syntactic information, most importantly the annotation of topological fields and of noun chunks with grammatical functions, is present with sufficient accuracy.

Chapter 6

Rule-based Approaches

6.1 The Resolution of Anaphora Procedure by Lappin and Leass

In chapter 3, we discussed the characteristics of rule-based approaches to anaphora resolution, and we presented an early representative of this class of resolution systems, Hobb's algorithm. Hobbs did not implement his algorithm as a running computer program, but simulated its behavior by hand.

Lappin and Leass (1994) later introduced a rule-based algorithm called "Resolution of Anaphora Procedure" (RAP) which they implemented as software. The authors state that they have applied the algorithm to both English and German. However, they only discuss the English variant in their paper. The algorithm resolves third person personal, possessive, and reflexive pronouns as well as reciprocals. To that end, Lappin and Leass combine several modules of two different types: Modules of the first type are designed as filters that rule out invalid combinations of pronouns and antecedents. Modules of the second type use linguistic rules to add and extend the information available to the resolution system about the pronouns and their candidate antecedents.

The modules are:

- A morphological filter that removes pairs that are morphologically incompatible from the candidate set.

- A module for identifying expletive pronouns. Expletives are excluded from further resolution.
- A syntactic filter that rules out pairs of a personal pronoun and a candidate antecedent that occur in the same local clause.
- An algorithm for selecting an antecedent for a reflexive or reciprocal within the same sentence.
- A procedure to compute salience values for an NP, based on a grammatical role hierarchy.
- A procedure for computing equivalence classes of coreferent NPs that are assigned the sum of the salience values of its elements.
- A final decision procedure which selects the highest ranked candidate as the antecedent for a pronoun.

In the following, we will discuss each module in greater detail.

6.1.1 The morphological filter

The morphological filter rules out pairs of pronouns and candidate antecedents that are not compatible with respect to their morphology, such as a plural pronoun and a singular antecedent. Since the functionality of this module is very similar to the author's own morphological prefilter, the reader is referred to the detailed discussion of the latter filter in chapter 8, section 8.1.

6.1.2 Test for expletive pronouns

For detecting expletive *it*, Lappin and Leass (1994) use two lists of modal adjectives and cognitive verbs. The list of modal adjectives contains elements such as

necessary	possible	certain	likely	important
good	useful	advisable	convenient	sufficient
economical	easy	desirable	difficult	legal

The class of cognitive verbs includes:

recommend think believe know anticipate assume expect

Given the two lists, Lappin and Leass define a set of patterns that indicate, if one of the patterns matches, that the pronoun is expletive:

It is **Modaladj** that **S**
It is **Modaladj** (for **NP**) to **VP**
It is **Cogv-ed** that **S**
It seems / appears / means / follows (that) **S**
NP makes / finds it **Modaladj** (for **NP**) to **VP**
It is time to **VP**
It is thanks to **NP** that **S**

6.1.3 The syntactic filter on personal pronouns

The purpose of the syntactic filter on personal pronouns is to remove pairs from the candidate set that are ruled out by binding principles. Lappin and Leass (1994) do not actually implement a full version of binding theory in their system. Instead, they introduce a number of configurational concepts which they use, together with the syntactic annotation created by their parser, to put constraints on personal pronouns and antecedents which capture a large amount of the binding principles. They define these concepts as follows:

Argument domain A phrase P is in the *argument domain* of a phrase N if and only if P and N are both arguments of the same head.

Adjunct domain A phrase P is in the *adjunct domain* of a phrase N if and only if N is an argument of a head H, P is the object of a preposition *PREP* and *PREP* is an adjunct of H.

NP domain A phrase P is in the *NP domain* of N if and only if N is the determiner of a noun Q and P is an argument of Q or P is the object of a preposition *PREP* and *PREP* is an adjunct of Q.

Containment A phrase P is *contained* in a phrase Q iff P is either *immediately contained* in Q, i.e. an argument or an adjunct of Q; or P is immediately contained in some phrase R, and R is contained in Q.

Based on these concepts, Lappin and Leass (1994) define six conditions that *rule out* coreference between a pronoun and an antecedent.

1. P and N have incompatible agreement features.
2. P is in the argument domain of N.
3. P is in the adjunct domain of N.
4. P is an argument of a head H, N is not a pronoun, and N is contained in H.
5. P is in the NP domain of N.
6. P is a determiner of a noun Q, and N is contained in Q.

Together, rules 2 to 6 specify the conditions under which a pronoun P is bound, in violation of binding principle B.

6.1.4 Antecedent selection for reflexives and reciprocals

Personal pronouns and reflexives/reciprocals are complementarily distributed. The set of rules on reflexives and reciprocals is structurally similar to the filters on personal pronouns. The difference is that while the rules on personal pronouns are implemented as *filters*, i.e. *remove* candidates, a candidate pair involving a reflexive/reciprocal is only added to the candidate set when it is *explicitly licensed* by the rules.

Lappin and Leass firstly impose a ranking on the relevant argument slots as follows:

1. surface subject
2. deep subject of a verb heading a passive VP
3. direct object
4. indirect object or PP object complement of a verb

The rules for selecting an antecedent are as follows. A noun phrase N is a possible antecedent for a reflexive or reciprocal A if one of the five rules below can be applied:

1. A is in the argument domain of N, and N fills a higher argument than A.
2. A is in the adjunct domain of N.

3. A is in the NP domain of N.

4. N is an argument of a verb V, there is an NP Q in the argument domain or the adjunct domain of N such that Q has no noun determiner, and (i) A is an argument of Q, or (ii) A is an argument of a preposition *PREP* and *PREP* is an adjunct of Q.

5. A is a determiner of a noun Q, and (i) Q is in the argument domain of N and N fills a higher argument slot than Q, or (ii) Q is in the adjunct domain of N.

6.1.5 Salience weighting

Alshawi (1987) states that salience is a property of the discourse context. The salience of an entity that occurs in a discourse does not remain constant over the time that the discourse evolves. Instead, it rises or drops according to the importance that the speaker or writer assigns the respective entity. The entity that is the most salient is the one that is currently focused in the discourse.

An NP in the candidate set of a pronoun that realizes a discourse referent that has high salience is likely to be an antecedent of the pronoun.

The salience weighting module in Lappin and Leass' system models the dynamic change in salience of the referents that are known to the system. It assigns each referent a number of *salience factors* that are determined by syntactic and positional properties of the NPs that correspond to the referents. The value of the individual salience factors is computed on the basis of a hierarchy (see table 6.1) which ranks the grammatical role and positional properties of the noun phrase. The system arrives at the final salience value by summing up the salience factor's values.

The rule systems discussed in the previous section apply only to pairs occurring within the same sentence, i.e. they formulate the additional restrictions on sentence anaphora. The salience principle is a discourse-wide phenomenon and therefore applies to both sentence and discourse anaphoric pairs.

6.1.6 Equivalence classes

In principle, it is possible to replace a noun phrase or a pronoun with any other noun phrase or pronoun if they both refer to the same entity. In more

Salience factor	Value
Positional salience factors	
Sentence recency	100
Salience factors based on grammatical role	
Subject emphasis	80
Existential emphasis	70
Accusative emphasis	50
Indirect object and oblique complement emphasis	40
Head noun emphasis	80
Non-adverbial emphasis	50

Table 6.1: Salience hierarchy used by RAP

formal terms, noun phrases and pronouns that refer to the same entity are referentially equivalent and therefore constitute an equivalence class. The number of noun phrases that are members of an equivalence class is equal to the numbers of times an entity is referred to in the discourse. With increasing frequency of mention, the salience of a referent increases as well. In Lappin and Leass' system, this discourse-dynamic, frequency-based increase of salience is reflected by replacing the salience value of an individual NP with the salience value of the equivalence class the NP belongs to, which is computed to be the sum of the salience values of its members.

6.1.7 Performance

Lappin and Leass evaluated their algorithm on a set of 345 sentences which were randomly selected from a 1.25 million word corpus of 48 computer manuals. The authors evaluated their system separately for intrasentential relations and relations spanning more than one sentence. On the 471 intrasentential cases, the system correctly resolved 403 pronouns, that is 86% accuracy.[1] There were only 89 cases of discourse anaphora (intersentential cases), here 72 relations were determined correctly (81% accuracy). In total, RAP correctly resolved 475 out of 560 instances (85% accuracy). These

[1] Lappin and Leass (1994) do not report separate figures for precision and recall, but only one measure for accuracy. This measure is related and equivalent to precision and success rate as discussed in chapter 4.

	Total	**Intersentential cases**	**Intrasentential cases**
Number of pronouns	560	89	471
Correctly resolved	475 (85%)	72 (81%)	403 (86%)

Table 6.2: Summary of results of the Resolution of Anaphora Procedure (Lappin and Leass, 1994).

results are summarized in table 6.2.

6.2 The knowledge-poor approach by Kennedy and Boguraev

Lappin and Leass' Resolution of Anaphora Procedure crucially relies on a deep syntactic analysis of the input data. If the approach is to be integrated in a complete end-to-end anaphora resolution system that is capable of resolving anaphora in unannotated text, it is vital that a sophisticated parser provides the anaphora resolver with syntactic structure of the necessary complexity and reliability. Although the performance of parsers has evolved substantially since the time when Lappin and Leass published their algorithm, it is still a non-trivial problem for a parser to deliver high performance in a domain independent environment.

The work by Kennedy and Boguraev (1996) addresses this issue. They devise a robust *knowledge-poor* algorithm for resolving anaphora that does not require input data annotated with complex syntactic structure, but is content with shallow morphosyntactic information about the tokens in the input text. Kennedy and Boguraev show that in spite of the much more restricted information, the performance of their knowledge-poor algorithm is comparable to that of the knowledge-rich approach by Lappin and Leass.

Kennedy and Boguraev's system ("knowledge-poor RAP", abbreviated KP-RAP henceforth) operates in two main stages. The first stage is the preparation of the unannotated input data, which includes the shallow syntactic and morphological analysis of the text. The second stage is the resolution proper. The following two sections will outline the system in more detail.

6.2.1 Input data

Instead of relying on a full parser, KP-RAP uses shallow morphosyntactic annotations added to the raw input text by the LINGSOFT morphosyntactic tagger. The annotation comprises the parts of speech, disambiguated morphological information, and information about the grammatical function of each token. Furthermore, the tagger outputs the word offset of each token in the text. The latter information is vital for the resolution approach: Using this offset, KP-RAP can compute precedence relations between tokens, which are used to shallowly determine coarguments.

The data such annotated is then post-processed with a module that recognizes and analyses partial constituents on the basis of a set of regular expressions that operate on the metatags added to the text by the tagger. The post-processor performs three main tasks:

1. Extraction of modifier-head sequences from noun phrases by means of a regular grammar that describes possible tag sequences.

2. Detection of subordinate environments. Again, this is accomplished only by means of a regular grammar that looks for sequences of tags that indicate subordinate structures. This way, the system finds noun phrases that are contained in adverbial adjuncts or in prepositional or clause complements of other NPs or relative clauses. Obviously, this task would be one that would normally be at the heart of the functionality of a parser. The regular grammar employed here is of course much weaker than a full-fledged parser, therefore, the accuracy delivered by this task cannot be expected to be equivalent to that of a parser.

3. Identification of expletive *it* by means of a set of context patterns, in a fashion similar to the extensive work by Paice and Husk (1987).

The NPs recognized this way are passed on as the markables to the downstream system.

6.2.2 Resolution

The resolution step performs the actual resolution of pronouns (both reflexive/reciprocal and personal pronouns) to their antecedents. In their

algorithm, Kennedy and Boguraev chose to assume a set model for representing coreference (in the sense of the classifications given in chapter 3). When two discourse referents are found to be coreferent, a unique coreference object is created that represents the common referential properties of the discourse referents. Pointers to this coreference object are added to both discourse referents. If additional coreferent discourse referents are found, they will be made to point to this coreference object as well. This structure is essentially an explicit representation of the set membership relation by means of pointers between the discourse referents and the coreference object.

The resolution algorithm covers all personal pronouns as well as all reflexive and reciprocal pronouns in the input text, in a way very similar to the RAP approach by Lappin and Leass (1994). The pronouns in the text are processed left to right. For each pronoun, a set of candidate antecedents is computed first. For reflexives and reciprocals it is required that the antecedent is a coargument of the pronoun. Since no deep syntactic analysis is available, coarguments are determined only on the basis of the precedence relation and the grammatical function annotation of each token. For the resolution of personal pronouns, the candidate set is run through two filters: A morphological filter removes candidates that are morphologically incompatible with the pronoun. The second filter is the disjoint reference filter. This filter may be called a "knowledge-poor implementation of binding theory", as its purpose is to make sure that any candidate antecedent satisfies the following three conditions relative to the pronoun:

Condition 1: A pronoun cannot corefer with a coargument.

Condition 2: A pronoun cannot corefer with a nonpronominal constituent which it both commands and precedes.

Condition 3: A pronoun cannot corefer with a constituent which contains it.

While in its functionality, the disjoint reference filter performs the same tasks as the binding theory filter in Lappin and Leass' approach, it again only relies on the precedence information available on tokens and the shallow annotation of embedding added by the subordination detector in the data preparation step.

The most important means of ranking candidate antecedents for selection is a salience hierarchy that is largely equivalent to that used by Lappin and Leass (1994). The salience of a discourse referent is computed in the same way as in RAP by summing over all individual salience factors, and then by dynamically updating the salience based on the distance of the discourse referent to the pronoun, and by the number of referents that are in the same coreference class. Furthermore, salience is affected by penalizing cataphoric constructions, while candidate pairs that occur in the same local context and pairs with a combination of grammatical functions seen previously receive a higher salience. The latter is to reward parallelism of grammatical functions. Unlike Lappin and Leass' parallelism reward, which is added to salience when the pronoun and the candidate have the same grammatical function, Kennedy and Boguraev's variant requires that the *pair* of grammatical functions of the pronoun and the candidate has been previously seen.

6.2.3 Discussion

Kennedy and Boguraev (1996) report an accuracy of 75% of their approach on a random selection of genres. This is 10 points of percentage lower than the accuracy of Lappin and Leass (1994). Kennedy and Boguraev argue that nevertheless the performance of their algorithm is comparable to that of RAP, since the latter was developed and evaluated in a closed domain of computer manuals, which is considered to be a fairly well structured text genre. Furthermore they find that the number of errors that can directly be attributed to the missing deep syntactic analysis is very small. They point out that the majority of problems can be resolved by improving the processing modules operating on the shallow annotation.

6.3 RAP for German

The following sections are dedicated to the author's re-implementation of Lappin and Leass' Resolution of Anaphora Procedure for German ("G-RAP"). The design of the system is closely modeled after Lappin and Leass' original English version and comprises three main components.

- **Morphological filter.** While Lappin and Leass check the agreement of a pronoun and potential antecedent in person, number, and gender within their syntactic filter, we use a dedicated separate module for prefiltering the candidate set. The morphological filter is shared by both G-RAP and the hybrid resolution approach to be discussed in chapter 8. We will describe the filter within the scope of the latter system.

- **Syntactic filter.** The syntactic filter puts binding constraints on pronouns and antecedents if they occur within the same sentence. Just as in RAP and KP-RAP, we do not implement a full configurational version of binding theory but rely on the notion of containment in coarguments (see section 6.3.4 in this chapter).

- **Salience weighting.** The salience weighting module computes salience values for all referents in the discourse on the basis of a salience hierarchy. We optimized the weighting of the individual salience factors in this hierarchy for German. We use the same strategy as Lappin and Leass for modeling the dynamic nature of salience. The module for computing salience is described in section 6.3.3.

6.3.1 Input data

We used the TüBa-D/Z treebank of newspaper text (see chapter 5) as our source data. We split the corpus in 1 188 individual articles containing 25 312 sentences.[2] This data set contains 13 278 personal, reflexive, and possessive pronouns in third person, which is the set of pronouns that our rule-based algorithm can resolve. Lappin and Leass consider the same set of (English) pronouns in their original implementation.

The referential relations that the pronouns in this set are members of are of three different types: *anaphoric*, *cataphoric*, and *bound*. Seven cases of *coreferential* relations are annotation errors. Additionally, the set contains expletive pronouns and pronouns that are in no anaphoric relation. The distribution of the relations is shown in table 6.3. As illustrated in figure

[2]Release 3 of the TüBa-D/Z treebank actually comprises 1 285 individual articles. However, the annotation of anaphora and coreference was an ongoing project at the time the experiments were conducted. Therefore, we excluded the unfinished 97 articles from our data set.

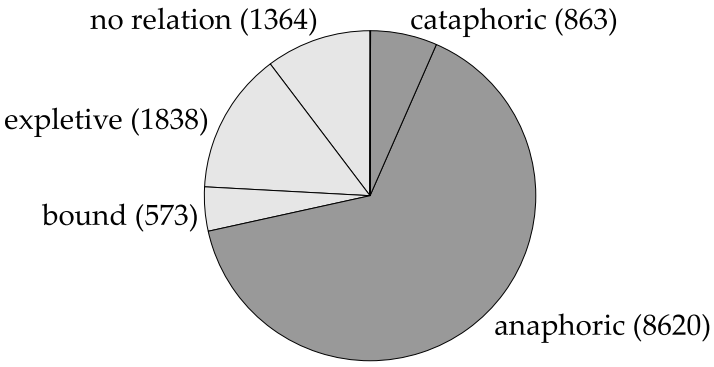


Figure 6.1: Distribution of referential relations of third person pronouns in the first 1 188 articles of the TüBa-D/Z treebank. The relations in dark gray are handled by G-RAP.

anaphoric	8 620
cataphoric	863
expletive	1 838
no relation	1 364
bound	573
coreferential	7

Table 6.3: Distribution of referential relations of third person pronouns in the first 1 188 articles of the TüBa-D/Z treebank.

6.1, we only handle the relations of *anaphora* and *cataphora*, which together constitute 9 438 of the relations (71%).

Expletive pronouns were removed from the set of candidates. Unlike Lappin and Leass, who use a pattern-based heuristics to detect expletive pronouns, we do not include an automatic mechanism but rely on the manual gold standard annotation in the TüBa-D/Z referential layer for the filtering.

6.3.2 Resolution algorithm

G-RAP's resolution procedure is functionally equivalent to the original RAP algorithm, but it was completely re-designed from scratch. Figure 6.2 on page 126 shows a flowchart. The input to the algorithm is the list of can-

didates, which is already morphologically filtered. Thus, the list only contains third person personal, reflexive, and possessive pronouns and their morphologically compatible potential antecedents. The list is ordered according to the sequence of occurrence of the pronouns in the text. The algorithm works through the list top to bottom, which is equivalent to a left to right, sentence by sentence, pronoun by pronoun order of processing. For each candidate, its dynamic salience value is calculated first, using the algorithm that is explained in section 6.3.3 and that is illustrated in figure 6.3 on page 127. Candidate pairs that occur in the same sentence must obey the binding principles, therefore pairs that violate the principles are removed from the candidate list in the next step. This filtering step is described in section 6.3.4. The resolution step proper is actually quite simple: The candidate with the highest salience is chosen as the antecedent. In case of a tie situation, the candidate that is closer to the pronoun is selected. If there are no candidates left for a pronoun after filtering, the algorithm moves on to the next pronoun. Otherwise, the pronoun and the antecedent are added to the same equivalence class.

6.3.3 Computing salience

The module for computing the salience of discourse referents has at its core a ranked hierarchy of salience factors. The hierarchy consists of a combination of factors depending on the grammatical functions of the NPs in the text, on their syntactic configuration, and on positional properties of a candidate with respect to a pronoun. Compared to RAP's original hierarchy, the hierarchy for G-RAP puts a slightly stronger emphasis on non-configurational salience factors. We determined the optimal values for the salience factors in an empirical process which we are going to describe later in this section. The salience factors are summarized in table 6.4.

G-RAP considers four salience factors that are based on the grammatical function of the potential antecedent which reflect the increasing obliqueness of the arguments:

- **Subject emphasis** (170 points, grammatical function `ON`)
  Discourse referents that are realized in the subject position receive a very high initial salience value. This value is substantially higher than all other salience factors, which reflects that subjects are usually highly salient entities in a discourse and frequently are the entities

that a pronoun refers to. Furthermore, the value is more than twice as high as the original value in Lappin and Leass's English implementation, which is only 80.

- **Accusative object emphasis** (70 points, grammatical function OA)
  Discourse referents that are realized in the position of the accusative object receive an initial salience value of 70. Although not as strong, this value is again higher than the corresponding value in RAP by 20 points.

- **Dative object emphasis** (50 points, grammatical function OD)
  Discourse referents that are realized in the position of the dative object receive an initial salience value of 50. This salience factor is related to Lappin and Leass' factor for indirect objects and oblique complements, which is set to only 40 points. While the latter does not further distinguish the syntactic entities, G-RAP includes distinct factors for dative and genitive.

- **Genitive object emphasis** (50 points, grammatical function OG)
  Discourse referents that are realized in the position of the genitive object receive an initial salience value of 50. This factor is not present in RAP. Although the German version supports in principle the distinct treatment of dative and genitive objects (unlike RAP), the optimal settings of the dative and genitive salience factors turned out to be equal. The dative and genitive factors together are thus functionally very similar to the indirect object and oblique complement factor in RAP.

The fifth salience factor **head noun emphasis** counts 80 points if

- the noun phrase α has the grammatical function HD and

  (a) α is not dominated by any other NP, *or*

  (b) α is dominated by one or more NPs, but no non-head NP or other category intervenes on the path between α and the topmost non-embedded NP.

  This salience factor is essentially equivalent to the corresponding factor in RAP.

Salience factor	G-RAP	RAP
Salience factors based on grammatical role		
Subject emphasis	170	80
Accusative object emphasis	70	50
Dative object emphasis	50	–
Genitive object emphasis	50	–
Indirect object and oblique complement emphasis	–	40
Parallelism reward	35	35
Further syntactic salience factors		
Head noun emphasis	80	80
Existential emphasis	–	70
Non-adverbial emphasis	–	50
Positional salience factors		
Short distance cataphora penalty	-80	–
Long distance cataphora penalty	-175	–
Cataphora penalty	–	-175
Current sentence reward	20	100

Table 6.4: Salience hierarchy used in G-RAP and comparison to RAP's hierarchy.

- the noun phrase α is a non-head NP (grammatical function "--") and is *not* embedded in any other NP. Differing from RAP, it was an empirical finding when optimizing the salience weights for constituents that *non-head, non-embedded* NPs should also receive increased salience.

The **parallelism reward** factor applies if the pronoun and the candidate antecedent have the same grammatical function. It is different from the salience factors discussed so far in a number of ways. Firstly, while the latter factors are mutually exclusive (because any noun phrase can only bear one grammatical function), the parallelism reward salience factor may apply in addition to one of the factors above. Furthermore, the parallelism reward depends on a *binary* property of the pronoun and the candidate antecedent, while the other salience factors are only determined by features of the individual NPs. We found the original value of 35 suitable for G-RAP

as well.

We did not include RAP's salience factors for existential emphasis and non-adverbial emphasis. The former uses a highly language-dependent pattern-based heuristics for existential constructions which does not apply to German. The latter factor rewards NPs that are not contained in adverbial PP adjuncts. This factor is captured by the head salience factor and the non-embedded, non-head salience factor.

The three positional salience factors are all binary factors. The **current sentence reward** (called "sentence recency" in RAP) increases the salience of candidates that occur in the same sentence as the pronoun by 20 points. The value of this reward is substantially lower than the corresponding value in RAP, which is 100.

The final two factors control, or rather penalize, cataphoric relations. They are significantly less frequent in discourse than anaphoric relations (see figure 6.1), and, from the perspective of the efficiency of information conveyance, suboptimal. G-RAP employs *two distinct* negative factors, called **short distance cataphora penalty** and **long distance cataphora penalty**. This is a notable deviation from the English original, which only uses one strong penalty factor of value -175 for all cataphoric relations. The short distance cataphora penalty is less strict (-80) and pertains to cataphoric configurations between a pronoun and an antecedent that occur in the same *local clause*:

(1) Nach der Kündigung **[seines]**$_i$ Mietvertrages hat **[ein**
After the cancellation his tenant agreement has a
Wiesbadener]$_i$ am Mittwoch abend seine Wohnung
man from Wiesbaden on Wednesday evening his apartment
angezündet.
set on fire.

'After he had received the cancellation of his tenant agreement, a man from Wiesbaden set his apartment on fire on Wednesday evening.'

The long distance cataphora penalty reduces the salience of an NP by 175 points (this value is equal to the one used by Lappin and Leass), and applies to configurations where the pronoun and the candidate are not located in the same clause,[3] configurations of the kind as illustrated in the

[3]This includes pairs that are located in the same *sentence*, but not the same *clause*.

following example. The possessive pronoun *ihr* in (2) refers to *der jungen Frau – the young woman*$_{\text{DAT}}$ in (3):

(2) Eigentlich sei das nicht **[ihr]**$_i$ Ding, hier mit dem weißen
Actually be that not her thing, here with the white
Umhang der Gewerkschaft Handel, Banken und Versicherungen
coat of the union of trade, banks, and insurances
(HBV) "im Regen rumzulatschen".
in rain tramp

'Actually it was not really up to her alley to "tramp the rain" wearing the white coat of the union.'

(3) **[Der jungen Frau]**$_i$ aus der Devisenabteilung
The young woman from the department of foreign currencies
einer Großbank, die unter dem "Duschvorhang" ein graues
of a large bank, who under the "shower curtain" a gray
Designerkostüm trägt, steht das Wasser in den Pumps.
designer costume wears, stands the water in the pumps.

'The pumps of the young lady from the department of foreign currencies of a large bank, who wears a gray designer costume under the "shower curtain", are soaked with water.'

Just like in the English original, the cataphora penalty is (apart from the distinction between short and long distance) a constant value that is subtracted from the salience. Since the salience of an NP is made to decay exponentially in dependence on the sentence distance, a constant reduction of salience in cataphoric configuration is sufficient and avoids overpenalization.

We determined the optimal values for the salience factors using an iterative empirical strategy. We started out with the values suggested by Lappin and Leass for RAP, and then decremented and later incremented the values in intervals of 10, leaving the other salience factors constant. The minimum value for each factor was 0, and the maximum value 200. This way, we arrived at the settings described.

The salience value for a referent is then computed as illustrated in figure 6.3. First, the static salience properties of a referent are determined on the basis of the salience hierarchy. Each factor that applies to the syntactic and positional properties of the referent is added to its salience. To determine the final salience value, two additional pieces of information that

represent the more dynamic discourse properties are incorporated in the calculation in analogy to RAP: The distance of the NP to the pronoun, and its membership in an equivalence class. The salience of the NP decreases with increasing distance to the current pronoun. In G-RAP, this is modeled by halving the salience with each intervening sentence: The salience of an NP that is located one sentence ahead of the pronoun is halved. The salience of an NP two sentences to the left of a pronoun is divided by four, and so on. In general, the salience S of an NP with a sentence distance of d to the pronoun and a salience value s is computed according to the formula:

$$S = \frac{s}{2^d}$$

Finally, all NPs that refer to the same entity are added to their equivalence class. Each equivalence class represents the accumulated salience of all of its members. The salience of the individual NPs is replaced with the salience of the whole equivalence class.

6.3.4 Candidate filtering

After determining the candidate antecedents for a pronoun and computing their respective salience values, a number of syntactic filters are applied in a fashion similar to the original implementation by Lappin and Leass for English. They only apply to pairs that are located in the same sentence:

- Personal pronouns *must not* be contained in the candidate antecedent's argument domain.
- Personal pronouns *must not* be contained in the candidate antecedent's adjunct domain.
- Reflexive pronouns *must* be contained in the candidate antecedent's argument domain, and the candidate must fill a higher argument slot.

The filters above depend on the terms argument domain and adjunct domain which have been adapted for G-RAP as follows.

Argument domain

An NP or pronoun n is in the argument domain of an NP or pronoun m iff

- n does not have the grammatical function `HD`
- n is a sibling of m and has one of the grammatical functions `ON`, `OA`, `OD`, `OG`.

Adjunct domain

An NP or pronoun n is in the adjunct domain of an NP or pronoun m iff

- n is contained in a prepositional phrase (i.e. the parent category p of n is `PX`), and m is a sibling of p with a grammatical function of `ON`, `OA`, `OD` or `OG`.

6.3.5 Resolution

From the filtered set of candidate antecedents, the candidate with the highest salience is picked as the antecedent of the pronoun. If two candidates have the same salience value, the candidate that is closer to the pronoun is selected as its antecedent.

6.3.6 Evaluation and discussion

We evaluated G-RAP by computing pairwise precision and recall. Since the system does not output full anaphoric chains, other strategies of evaluation are not applicable. We counted an antecedent as correct if the NP selected by G-RAP had the same head as the gold-standard antecedent. In this configuration, the G-RAP system achieved precision of 76.6% and recall of 76.5%, resulting in the f-measure of 76.6%. Table 6.5 compares the performance of the three rule-based systems discussed in this chapter. Both RAP (Lappin and Leass, 1994) and KB-RAP (Kennedy and Boguraev, 1996) do not report separate figures for precision and recall, but only one value for accuracy, or correctness, which is essentially equivalent to precision. In comparison to RAP, G-RAP performs 10 points of percentage worse. G-RAP performs slightly better than KB-RAP: KB-RAP achieves 75%, and G-RAP reaches 76.6%. Thus, KB-RAP and G-RAP perform within the same range. Kennedy and Boguraev's argument that the weaker performance of

Approach	Precision	Recall	F-measure
RAP	85%	–	–
KB-RAP	75%	–	–
G-RAP	76.6%	76.5%	76.6%

Table 6.5: Results of RAP, KB-RAP, and G-RAP

their system is due to the fact that they use more data with higher variability is applicable to G-RAP as well.

The comparison of the salience hierarchies for the English and the German variants for RAP shows that features that pertain to position get lower weights in the German version, and features that pertain to grammatical function get higher weights:

The most notable effect concerns the "current sentence reward", which increases the salience of a potential antecedent if it occurs in the same clause as the pronoun. We experimented with both decreased and increased weights. In both settings, the performance of the system is lowered. However, the loss is notably larger with higher weights.

Potential cataphoric relations are penalized in RAP, since they occur less frequently in discourse than anaphoric relations. In the German version, performance improves when this strategy is relaxed and a distinction between "short distance" cataphora and "long distance" cataphora is introduced: Potential postcedents that occur within the same sentence as the pronoun are less penalized than potential postcedents that occur in following sentences.

Both the current sentence reward and the cataphora penalty are based on positional features – the relative position of the pronoun and a potential antecedent. Lappin and Leass (1994) optimized their weights on a corpus of computer manuals. This is a text genre where specific attention is paid to establishing highly coherent, step-by-step text with a relatively simple discourse structure. Newspaper text on the other hand, which is the data source that we used in our experiments with G-RAP, is a genre which is known to be stylistically more variable, including the locations when open referential relations are resolved in the text.

Thus, although the re-implementation of RAP was initially only carried out as a basis of comparison for the hybrid approach to be discussed in

chapter 8, the system provided interesting results in its own right. Apart from the insights proper, this illustrates the advantage of the possibility of directly inspecting and interpreting the rules in a white-box system.

A further concluding remark seems in order about the nature of the resolution approaches that we discussed in this chapter. We characterized them as rule-based approaches, clearly distinguishing them from data-driven approaches. However, this distinction is actually not as clear-cut: Comparing Hobb's algorithm with the implementations of RAP, we find that an important new concept in RAP is the ranked hierarchy of salience factors, which is essentially a set of hand-tuned linguistic features. While a linguistic rule is an explicit formulation of some requirement (such as the path configurations in Hobbs' algorithm), a feature is just a representation of a relevant property, requiring an additional decision function that checks the features. In RAP, this decision function is the selection procedure that picks the candidate with the highest final salience value. Thus RAP occupies a middle position between plain rule-based approaches and plain data-driven approaches which *only* rely on features. The difference to the latter systems is that in RAP, the features are not extracted automatically, but determined manually, by an expert linguist. We will retain our categorization of RAP as a rule-based approach, since the distinction between manually formulating resolution principles based on linguistic rules and the automatic extraction of features in data-driven approaches seems most important to us. However, we suggest the term *approaches based on expert knowledge* as an alternative for systems of this kind.

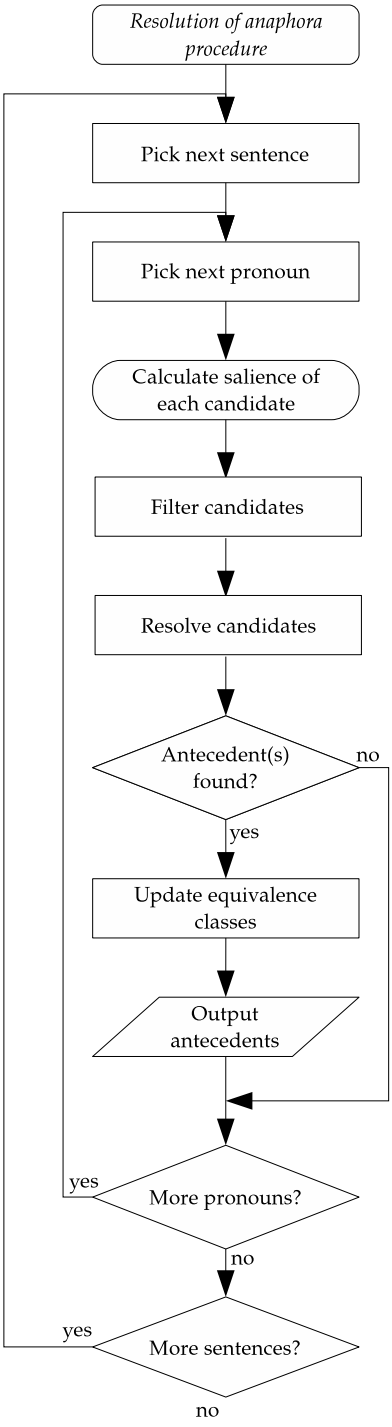


Figure 6.2: Flow chart of German RAP

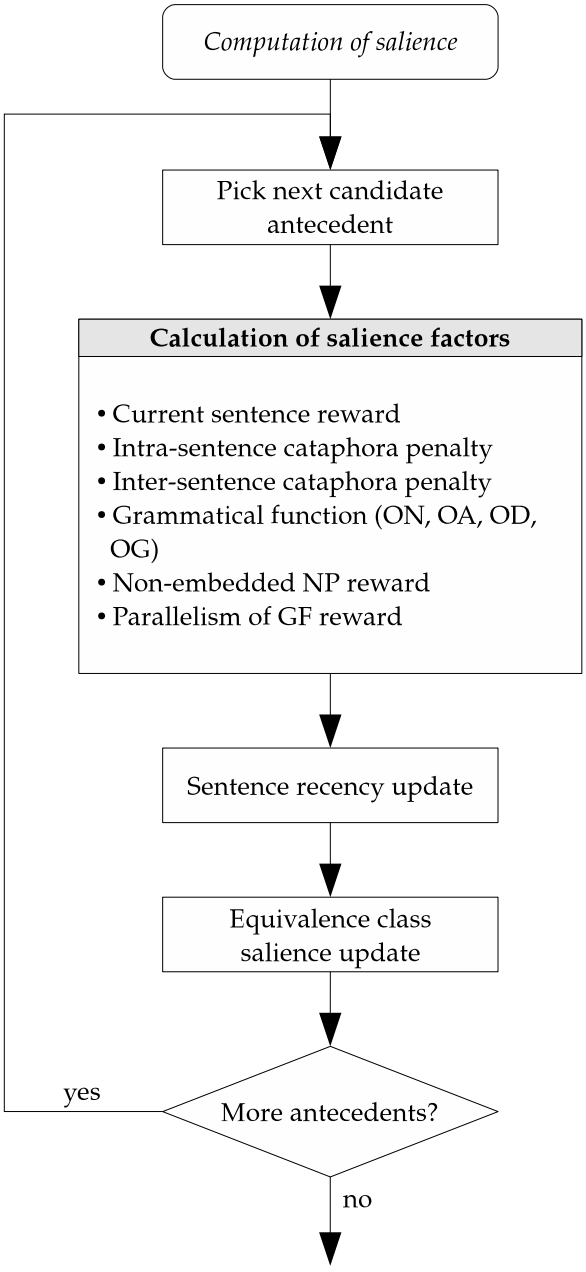


Figure 6.3: Computation of salience values

Chapter 7

Machine-learning-based Approaches

As discussed in detail in chapter 3, approaches to anaphora resolution based on machine learning strategies do not require a sophisticated system of linguistic rules. Instead, they autonomously extract their own linguistic model from training data.

In this chapter, we will discuss several approaches to anaphora resolution that are based on machine learning techniques.

7.1 The decision tree based approach by Soon et al.

Soon et al. (2001) devise a knowledge-poor end-to-end anaphora resolution system based on a machine learning approach. Similarly to the rule-based system by Kennedy and Boguraev (1996), which we discussed in chapter 6, it does not require input data that contains comprehensive sophisticated linguistic analyses, but can do with comparatively shallow linguistic information. It is an "end-to-end" resolution system since it is capable to operate on raw text, i.e. it comprises all necessary modules to annotate the raw text, identify markables, and then finally resolve the relevant markables to their antecedents. In this respect also, the system by Soon et al. is very similar to Kennedy and Boguraev's, and many other approaches.

The significant difference however is the architecture of the core resolution module. While Kennedy and Boguraev's system includes a rule-based approach, Soon et al. implement a machine learning architecture based on

the decision tree algorithm C5, which is an extended version of the original C4.5 decision tree learning algorithm (Quinlan, 1993).

In accordance with many other resolution systems, the approach by Soon et al. operates in two stages. The first stage prepares the raw input text, i.e. annotates the text with the necessary linguistic information required by the second step, which performs the resolution proper. In what follows, we will describe the two stages in greater detail.

7.1.1 Data preparation and determination of markables

The first step in Soon et al.'s system is the data preparation step. It consists of a pipeline of linguistic processing modules which are applied in sequence to perform the following tasks:

1. Tokenization / sentence segmentation
2. Morphological processing
3. POS tagging
4. Identification of NP boundaries
5. Named entity recognition
6. Nested noun phrase extraction
7. Semantic class determination

POS tagging, NP boundary identification, and named entity recognition are carried out using statistical taggers based on Hidden Markov Models. For POS tagging, Soon et al. use the tagger by Church (1988), and for NE recognition, they employ the system by Bikel et al. (1999). In step 6, nested noun phrases are extracted. Nested noun phrases are either contained in possessive noun phrases (as in '*his* long-range strategy', or '*Eastern*'s parent), or modifier nouns, as in '*wage* reductions', or '*Union* representatives'. The set of markables that is considered by the resolution algorithm consists of all noun phrases, named entities, and nested noun phrases. Finally, noun phrases that were not assigned a named entity type by the NE recognition module are run through the semantic class determination module which attempts to assign a semantic class to the markable by looking it up in WordNet.

7.1.2 Features

Soon et al. (2001) use a set of twelve features. Of these twelve features, eight are binary features, i.e. features that represent properties pertaining to a pair of markables *i* and *j* that is currently processed. The remaining four features encode properties of single markables. The following table gives a description of all features.

Feature	Arity	Description
Features based on distance		
DIST	binary	distance between *i* and *j*
Features based on morphology		
NUMBER	binary	agreement of *i* and *j* in number
GENDER	binary	agreement of *i* and *j* in gender
Features based on syntax		
I_PRONOUN	unary	*i* is a pronoun
J_PRONOUN	unary	*j* is a pronoun
DEF_NP	unary	*j* is a definite NP
DEM_NP	unary	*j* is a demonstrative NP
APPOSITIVE	binary	*j* is an apposition to *i*
Features based on lexical information		
STR_MATCH	binary	string match of *i* and *j*
Features based on (shallow) semantics		
SEMCLASS	binary	semantic class agreement
PROPER_NAME	binary	*i* and *j* are proper nouns
ALIAS	binary	*i* is an alias of *j*

Table 7.1: Features used by Soon et al. (2001).

7.1.3 Generation of training data

The training data for the C5 decision tree classifier is extracted from the MUC-6 (MUC-6, 1995) and MUC-7 (MUC-7, 1997) training corpora, which are annotated manually. For any pair of markables *i* and *j* that is coreferent, a positive training sample is extracted. Negative training samples are generated such that for any *positive* instance of a pair *i* and *j*, all the *intervening negative* instances between *i* and *j* are extracted. With this ap-

proach, Soon et al. avoid for the frequency distribution of positive and negative training samples to become too skewed towards the negative samples, which would be the case when *all* negative instances of a pronoun were extracted.

Given the training data, the classifier constructs a decision tree. Decision trees are representations of a classification process that consists of a sequence of atomic decisions that finally lead up to the assignment of a class label. Each node in the tree corresponds to one atomic decision. The nodes are labeled with the features. Thus, any decision amounts to evaluating one feature and, depending on the value of that feature, moving on to the next node. The construction of the decision tree is the core task of the C5 classifier, where most of the work is to be done. Classifying new instances just involves fairly easy feature-checking.

7.1.4 Evaluation

Soon et al. (2001) report a recall of 58.6% and precision of 67.3% of their system on the MUC-6 data, yielding an f-measure of 0.626. On MUC-7, the system performs slightly worse, with recall at 56.1%, and precision at 65.5%, with an f-measure of 0.604.

As part of the evaluation of their system, Soon et al. thoroughly inspected the contribution of each of the twelve feature to the performance of the resolution system. They find when training the classifier only on one feature at a time, three features yield a nonzero performance: ALIAS, STR_MATCH, and APPOSITIVE, with the ALIAS and STR_MATCH features delivering f-measures substantially higher than the APPOSITIVE feature. It is interesting that both of these two features are features that represent lexico-semantic properties of the markables, albeit rather shallow, of course, whereas the other features which represent linguistic information traditionally used in algorithms for anaphora resolution obviously are much weaker. In fact, Soon et al. (2001) find that the difference in f-measure when only using these three features as opposed to using all features only amounts to 2.3%. This shows that, especially for the resolution of definite NPs, whose surface form is a strong indicator for their meaning, features that represent shallow semantics are vital for successful resolution.

7.2 The competition-learning approach by Yang et al.

The majority of approaches to pronoun resolution, including the one by Soon et al. (2001), select antecedents based on a pairwise model. As described in chapter 3 on resolution strategies, it is characteristic of pairwise models that the resolver considers only a pronoun and *one* candidate antecedent at a time. These pairs are ranked by solely relying on information about the pronoun and the candidate. Finally, the candidate that was ranked highest is selected as the antecedent of the pronoun. An alternative strategy is to conceive the resolution process as a *competition* between multiple candidates, as outlined in section 3.3.2 of chapter 3. In such a setting, the resolver considers multiple candidates *at the same time*, such that the properties of one antecedent have a direct influence on the ranking of the others. We find such a competition model in the rule-based implementations of the English and German RAP algorithms (see chapter 6). A machine-learning-based approach that adopts a competition model is the system by Yang et al. (2003). They train a C5 classifier (Quinlan, 1993) on *triples* consisting of one pronoun and two candidate antecedents. Thus, unlike pairwise models, they aim to represent the mutual influence of two competing candidates in the instances that are presented to the classifier. The triples are of the form $(C_i, C_j, P), i > j$, where P is the pronoun, and C_i and C_j are the candidate antecedents. C_i is always closer to the pronoun. Yang et al. use two types of samples:

- *positive samples*, where C_i is a positive candidate, and C_j is a negative candidate
- *negative samples*, where C_i is a negative candidate, and C_j is a positive candidate

For any pronoun, the complete set of instances consists of a number of positive samples and a number of negative instances. During classification, new triples are either assigned the positive or negative class. The final winner is determined by a post-processing step which scores candidates: If a triple (C_i, C_j, P) is classified as positive, then the score of C_i is increased (since C_i is the positive candidate in the pair). If the triple is classified as negative, then C_j's score is increased, because C_j is the positive candidate.

	MUC-6			MUC-7		
	R	**P**	**F**	**R**	**P**	**F**
Strube (1998)	75.4	73.8	74.6	58.9	56.8	57.8
Ng and Cardie (2002)	76.1	74.3	75.1	62.9	60.3	61.6
Conolly et al. (1997)	57.2	57.2	57.2	50.1	50.1	50.1
Yang et al. (2003)	**79.3**	**77.5**	**78.3**	**64.4**	**62.1**	**63.2**

Table 7.2: Results of Yang et al.'s competition approach on pronoun resolution

The candidate which got the highest score of all candidates is finally selected as the antecedent.

7.2.1 Evaluation

Yang et al. (2003) evaluate their approach on the MUC-6 (1995) and MUC-7 (1997) coreference data sets, and compare their approach to three other systems, the S-list algorithm by Strube (1998), the machine learning approach by Ng and Cardie (2002), and the approach by Conolly et al. (1997). Strube's algorithm is based on a variant of centering theory, Ng and Cardie's system is an extension of the decision tree based resolver by Soon et al. Conolly et al.'s system employs also a twin-candidate model, but unlike in Yang et al., all NPs that precede a pronoun are considered as candidates. The results are reproduced in table 7.2. The table shows that Yang et al. (2003) reach consistently better results than the other approaches. The greatest difference in performance is between the two approaches that employ competition models. This indicates that while the explicit representation of the competition of candidates is a useful piece of information, the choice of the other features considered in the resolution process is more important on the final result.

7.3 Memory-based learning

The paradigm of learning that is pursued by many approaches to machine learning is that of *learning by abstraction:* During the training phase, the learning component of the classifier builds an internal model from the training data that is of sufficient generality such that it applies to and is

capable of handling new data with as much accuracy as possible. Thus, the major part of the work is done during the training phase. The classification of new data instances just amounts to matching them against the general model, a task that requires fairly minimal processing effort. This strategy of learning is also frequently termed *eager learning*, reflecting the fact that most of the "hard" work is done during the learning phase.

The concept of *lazy learning* that underlies memory-based learning is fundamentally different: Lazy learning means that the processing load dedicated to learning is minimized, and the learning component effectively *avoids* to generalize over the training samples it sees, while the work load is transferred into the classification phase. In memory-based learning, the lazy learning paradigm is implemented as follows: Training *only involves storing* all training samples in memory – without any further modification or abstraction. In the classification phase, the classifier computes for each new sample its *similarity* to the samples previously stored, and assigns to the new sample the class of the in-memory sample that is most similar. Thus, a memory-based classifier consists of a *memory-based training component* and a *similarity-based classification component*.

Daelemans and van den Bosch (2005) argue in the introduction to their book on memory-based language processing that such a setup is especially advantageous for use with natural language processing tasks. It is characteristic for natural language that on the one hand the mechanisms that underlie its generation are highly regular. But on the other hand, language is full of exceptions. This poses a challenge to approaches that aim to deal with linguistic phenomena using machine learning strategies: A high degree of generalization will enable the system to represent the underlying regularities to a large scale, but it will be sensitive to exceptions, and, in the worst case, probably not be able to handle them at all. A system that is tailored to deal with all the exceptions is likely to fall victim to effects of overfitting the training data, resulting in poor performance of handling new instances of regular data.

Memory-based learning strikes a good balance between these two extremes, which is a result of the organization of the stored training samples. If one figures the entirety of all training samples as spanning a sample space, the samples will not end up distributed evenly throughout this space, but there will be agglomerations of samples in locations that cor-

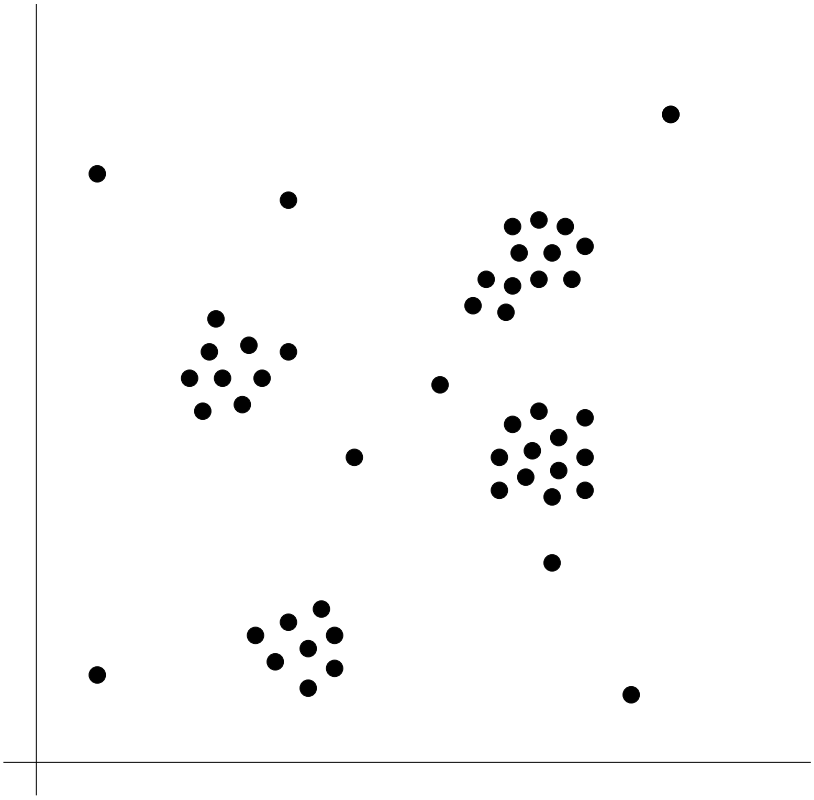


Figure 7.1: Schematic structure of a sample space in memory-based learning.

respond to regular linguistic phenomena (see figure 7.1). This is due to the fact that the majority of all samples are generated by regular linguistic processes yielding similar concrete instantiations of the phenomenon. Scattered in the space between these sample clusters, there will be additional samples that correspond to exceptions in the data. It is in fact the absence of any explicit attempt of generalization that leads to the balancing power of the memory-based approach: A new sample to be classified that corresponds to regular data will be very close to the samples in one of the clusters. Due to the high *average* similarity of the new sample to all samples in the cluster, the classifier will be able to recognize that the new samples belongs to that class. In other words, memory-based learning achieves generalization by computing average similarities. Exceptions in the data, on the other hand, will be most similar to one of the *singleton*

samples in the sample space corresponding to a similar exceptional case. Again, the classifier will assign this class to the new sample. Both tasks can be accomplished with the same one computation - the determination of the *nearest (or most similar) neighbors* of the new sample. This way memory-based learning is capable of both generalizing over the input data and at the same time preserving the ability to handle special cases.

To summarize, the classification strategy adopted by memory-based learning techniques is to take the training samples as prototypical examples of their respective classes, and judge new samples by their similarity to the training samples. In this respect, the approach is appealingly related to findings in cognitive psychology about the way humans represent categories: Instead of relying on rules or probabilistic reasoning, people compare new stimuli with the individual examples of categories they know, and decide about the category of the new stimulus by the grade of similarity of the new stimulus with the in-memory example. Eleanor Rosch showed in 1973 that people estimate class membership on a graded scale, with some members regarded more typical and other less typical while still belonging to this class: *Robin* was ranked 1.1 on a typicality scale between 1 (very typical) and 7 (very untypical) as a representative for the class of *birds*, while *chicken* ranked 3.8. *Murder* was regarded as very typical crime (1.0), while *vagabondage* wasn't (5.3). Thus people assign categories by the similarity of new stimuli to the prototypical examples they know. The way memory-based learning systems classify new samples is inspired by these cognitive strategies of categorization.

In the remainder of this section, we will describe in greater detail the k-nearest-neighbors algorithm, which is the underlying algorithm used in memory-based learning to classify new samples by similarity.

7.3.1 The k-nearest-neighbors algorithm

As mentioned before, the memory-based concept is an instance of lazy learning, which means that the computationally complex tasks are only carried out while classifying new samples of data instead of in the learning phase. In memory-based learning, this task is to determine the similarity between a new sample and the samples stored in the training phase. This algorithm is called the *k-nearest-neighbors algorithm*.

In memory-based learning, each sample is represented by a vector of

features that encode the relevant properties of the classes to be dealt with. The entirety of a all feature vectors spans a vector space whose dimensionality is equal to the number of features. Similarity in this vector space is defined in terms of the distance between two feature vectors. The smaller the distance between two feature vectors, the greater their similarity, and vice versa.

The basic functionality of the k-nearest-neighbors algorithm is quite simple: It determines the class of a new sample by inspecting the classes of the k feature vectors that are closest to the vector of the new sample, and then assigns the new sample the class that the majority of these k vectors belong to.

TiMBL's implementation

TiMBL, the Tilburg Memory Based Learner (Daelemans et al., 2005), implements a slightly modified version of the k-nearest neighbors algorithm. Instead of only considering the k *individual nearest vectors*, the classifier takes into account all feature vectors with the k *closest distances*, as illustrated in figure 7.2. The gray dot in the middle of the diagram represents the feature vector that corresponds to the sample to be classified. Here, all feature vectors are considered with the $k = 3$ closest distances. There are four vectors that are closest to the new sample, seven that are second-to-closest, and finally five with the third-to-closest distance, which amounts to a total of 16 feature vectors that are considered for determining the class of the new sample. Other feature vectors that are further apart are ignored.

Distance metrics

The crucial part in the implementation of the k-nearest neighbors algorithm[1] is the metric that is chosen to compute the distance between two feature vectors, as it turns out that the choice of the distance metric can substantially influence the performance of the classifier. Any distance metric d between two vectors x and y to be used by the classifier must satisfy the basic mathematical axioms for metrics:

[1] We will call the algorithm k-nearest *neighbors* in what follows, even though it should strictly speaking be k-nearest *distances*.

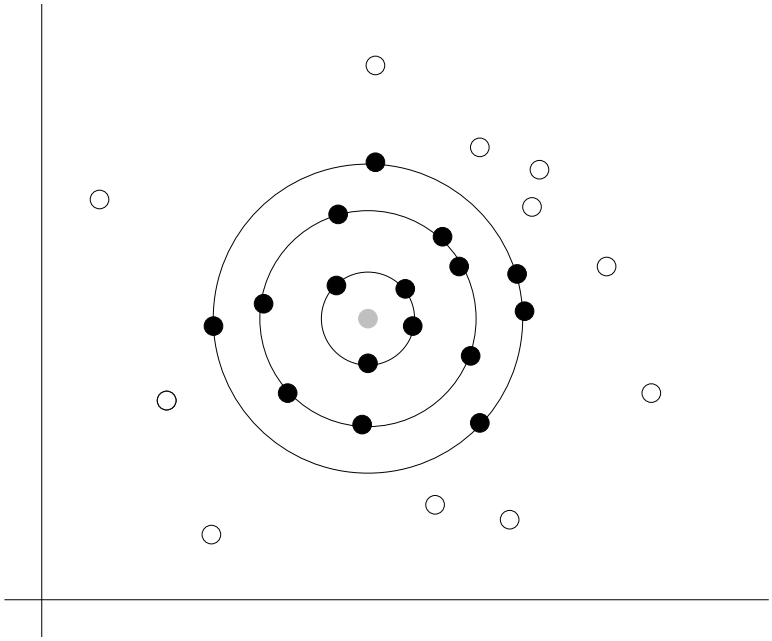


Figure 7.2: Feature vectors observed for the k nearest distances with $k = 3$. The vector of the sample to be classified is depicted by a gray dot, the feature vectors considered are black dots, and feature vectors ignored (due to their greater distance) are white.

1. **Definiteness**
   $d(x, y) = 0 \Leftrightarrow x = y$
   Definiteness states that if the distance between two vectors x and y is 0, then they must be identical.

2. **Symmetry**
   $d(x, y) = d(y, x)$
   Symmetry states that the distance between x and y is the same no matter whether measuring starts at x or at y.

3. **Triangle inequality**
   $d(x, y) \leq d(x, z) + d(z, y)$
   The triangle inequality states that the direct distance between two vectors x and y is never larger than the sum of the distances between x and an additional vector z, and z and y.

In addition to these axioms, in order to be useful in an NLP setting, a metric should meet some more requirements. Since NLP genuinely deals with language material, many of the features will be non-numeric, and rather be strings, such as words, POS tags, grammatical function tags, labels of anaphoric relations and so on. Furthermore, the features in a vector will likely not all be of the same data type – typically, numeric values and strings will alternate. Therefore, a distance metric should be able to deal with heterogeneous data in the feature vectors. The distance metrics that are implemented in the TiMBL classifier are specifically tailored to the requirements just described. We will now discuss a subset of the distance metrics supported by TiMBL in what follows.[2]

Overlap metric

The overlap metric is the simplest metric provided by TiMBL. It is a symbolic metric, i.e. it deals with both numeric and non-numeric features. The distance Δ between two samples X and Y is defined as:

$$\Delta(X,Y) := \sum_{i=1}^{n} \delta(x_i, y_i)$$

where

$$\delta(x_i, y_i) := \begin{cases} abs(\frac{x_i - y_i}{max_i - min_i}) & \text{if numeric, else} \\ 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases}$$

Thus, the overlap metric is a feature-wise metric, with the total distance being the sum of the distances of the individual features. For numeric features, the distance is computed as the difference of the feature values x_i and y_i, normalized by the range. For symbolic features, the distance simply is set to 1 if they are not equal, and 0, if they are.

It is obvious that all features are treated exactly the same, which is frequently undesirable. Therefore, a weighting factor w_i may be introduced for each feature that defines how much that feature contributes to the total distance, yielding the extended formula for overlap metric:

$$\Delta(X,Y) := \sum_{i=1}^{n} \mathbf{w_i} \delta(x_i, y_i)$$

[2]The reader is referred to the TiMBL manual (Daelemans et al., 2005) for a detailed description of all distance metrics supported by TiMBL.

TiMBL provides several strategies to determining the weighting factor w_i for each feature, which are based on information theoretic considerations. TiMBL's default feature weighting approach is the Information Gain weighting. Information Gain measures for each feature *separately* how much this feature contributes to the knowledge about a specific class. Mathematically, Information Gain is a probability-weighted average of the informativity of the feature values, and computed as the difference of the entropy of a class without taking into account the values of the feature and the entropy with knowledge about the feature values:

$$\mathbf{w_i} = H(C) - \sum_{v \in V_i} P(v) \times H(C \mid v)$$

where V_i is the set of all possible values of the feature, and

$$H(C) = -\sum_{c \in C} P(c) \log_2 P(c)$$

the entropy of the class labels.

A disadvantage of Information Gain is that it tends to overestimate features with many different values. Daelemans et al. (2005) give as example a database of hospital patients where one of the features is a unique patient ID. This feature has a very high information gain (the entropies $H(C \mid v)$ are very low, thus the sum $\sum_{v \in V_i} P(v) \times H(C \mid v)$ is small as well, yielding high values for w_i).

Therefore Quinlan (1993) suggests a normalized version of the Information Gain measure, called Gain Ratio, which is the ratio of the Information Gain of a feature divided by split info $si(v)$, the entropy of the values of the feature:

$$\mathbf{w_i} = \frac{H(C) - \sum_{v \in V_i} P(v) \times H(C \mid v)}{si(i)}$$

where

$$si(i) = -\sum_{v \in V_i} P(v) \times \log_2 P(v)$$

For features with many values, the split info is high as well, thus the fraction w_i becomes smaller. By introducing the normalization factor of split info, Gain Ratio thus reduces the overweight of many-valued features.

TiMBL supports further weighting schemes whose discussion is beyond of this chapter's scope. The reader is referred to Daelemans et al. (2005) for additional information.

For symbolic features, the overlap metric introduced in the previous section yields rather coarse results. This is because for features of this type, it can only check for exact equality, or inequality. As Daelemans et al. (2005) put it *"this means that all values of a feature are seen as equally dissimilar"*. For example, a feature that encodes POS tags might, among others, take the values `NN`, `NE`, and `VVFIN`, standing for proper nouns, named entities, and finite verbs in the German STTS tagset.[3] Overlap metric, limited to comparing symbolic feature values, would treat two vectors with values `NE` and `VVFIN` for the POS feature as equally dissimilar to a vector with value `NN`, even though the vector with the feature value `NE` is certainly much more similar to the one with feature value `NN` than the vector with `VVFIN`. One possibility to solve this problem would be to define specialized domain-dependent operators for the overlap metric that are able to properly determine equality for specific kinds of features. Obviously, this solution would not be sufficiently general, as for any new type of feature, new comparison operators would have to be defined, programmed and integrated with the existing system. The Modified Value Difference Metric (MVDM), which we will discuss in the next section takes a different approach to solve this problem.

Modified Value Difference Metric

The Modified Value Difference Metric (MVDM, Cost and Salzberg 1993) is a metric for computing distances between features in a way independent of the feature's domain. MVDM does not consider at all the concrete values of a feature, but evaluates the difference in influence of different values of the same feature on the probability that the vector with that specific value is a member of a class C_i. Coming back to the simple example from the previous section, if the class C_i encoded types of phrases, and the feature vectors encoded properties of the heads of these phrases, feature vectors with values of `NN` and `NE` are very likely to belong to a class NP, while a feature vector containing the value `VVFIN` is likely to belong to VP. This difference in probability of class membership can be exploited to define a

[3] We discussed the Stuttgart-Tübingen Tagset in chapter 5. The full tagset is listed in appendix A.

distance metric:

$$\delta(v_1, v_2) := \sum_{i=1}^{n} |P(C_i|v_1) - P(C_i|v_2)|$$

The above formula is the definition of the MVDM. It is the sum of the individual differences in class membership over all classes given two different values of the same feature v_1 and v_2.

Distance-weighted class voting

As mentioned earlier, the k-nearest-neighbor algorithm assigns a class to a new sample essentially by observing the majority vote of the classes of the k nearest neighbors of the new sample. Thereby, the vote of each of the k neighbors weighs equally. However, as Daelemans et al. (2005) argue, this may pose a problem with very sparse data. In this case, the number of instances found within the k nearest distances is typically quite low, and therefore, the predictive power of the model is limited. A sensible approach to remedy this is to increase k to consider more samples in the instance base. However, the similarity of these additional samples to the core instance decreases with increasing distance. Treating all votes the same might therefore actually deteriorate the classification, since samples further apart are worse indicators for the correct class than closer samples.

A solution for this problem is to introduce weights on the votes depending on the distance band the respective vector is located on.

The simplest weighting function is the *inverse linear distance weight*, which linearly reduces the weights on the votes with increasing distance:

$$w_j = \begin{cases} \frac{d_k - d_j}{d_k - d_1} & \text{if } d_k \neq d_1 \\ 1 & \text{if } d_k = d_1 \end{cases}$$

If the distance is $k = 1$ (i.e. closest possible), the weight is defined to be 1. The weights for all other distances are linearly distributed over the minimum and maximum distance.

Inverse distance weight non-linearly reduces the weight, defined as follows:

$$w_j = \frac{1}{d_j} \text{ if } d_j \neq 0$$

The weight w_j of the vote of neighbor j is simply the reciprocal of its distance d_j.

A weighting function that is inspired by findings in cognitive psychology that the salience of a stimulus decays exponentially (Shepard, 1987) can be defined as follows:

$$w_j = e^{-\alpha d_j^{\beta}}$$

where α and β are variables that control the speed and shape of the decay, as illustrated in figure 7.3.

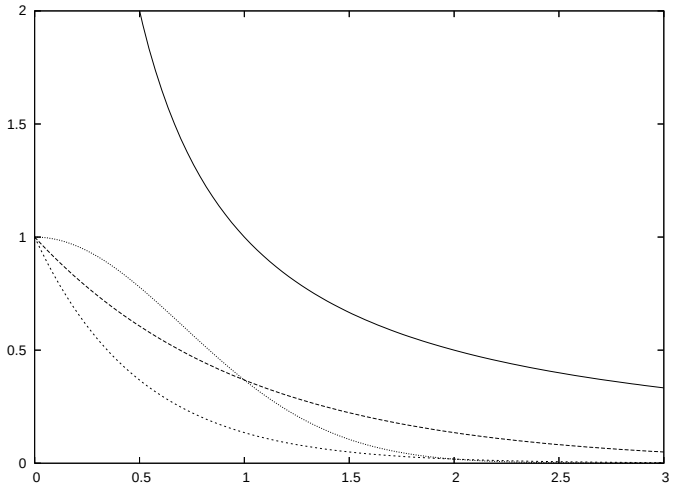


Figure 7.3: Decay functions. Top to bottom: inverse distance, exponential decay with $\alpha = 1, \beta = 1; \alpha = 2, \beta = 1; \alpha = 1, \beta = 2$.

7.4 The memory-based approach by Preiss

Judita Preiss (2002) examines whether a machine-learning-based approach is capable of achieving comparable levels of performance as a rule-based system when provided with features that are equivalent to those used in the rules of a rule-based system. Rule-based systems such as RAP (Lappin and Leass, 1994) or the knowledge-poor approach by Kennedy and Boguraev (1996) require a careful, manual weighting of the features, which is a labor-intensive process. An example for this is the salience hierarchy that is employed in both Lappin and Leass' and Kennedy and Boguraev's systems where the features that pertain to the computation of salience factors had to be empirically ranked by the authors. Preiss replaces this manual step of ranking salience features with a

memory-based machine learning algorithm, and examines to what extent the learning algorithm can infer the correct model parameters.

7.4.1 System architecture

Preiss' system consists of a preprocessing module for preparing raw text, and the resolution module proper. For the former, she integrates the preprocessor from her re-implementation of Kennedy and Boguraev's system, which annotates the text with markable boundaries and grammatical function information.

The resolution module proper is based on the Tilburg Memory Based Learner (TiMBL; Daelemans et al. 2005). As with all machine learning systems, TiMBL requires a training phase in which it computes its internal model. Once the model has been acquired, TiMBL is ready for classifying new samples in the classification (or testing) phase.

As a training corpus for the TiMBL classifier, Preiss prepared the first 2401 sentences of the first corpus of the written section of the BNC by manually annotating anaphoric relations between third person personal and reflexive pronouns and their antecedents. For each pair of a pronoun and an antecedent, a training sample is extracted which consists of a feature vector that represents the relevant linguistic information about the pair (the features are explained in detail in the next section).

Unseen raw text whose pronouns are to be resolved is run through the Kennedy and Boguraev preprocessor, which annotates the text with markable boundaries and grammatical function information. Preiss' reimplementation lacks a morphological filter component, but instead of using the LINGSOFT tagger, she employs the parser of Briscoe and Carroll (1993), which delivers both full parses (if possible) as well as partial parses and grammatical functions. Based on this syntactic analysis, the system extracts the feature vectors for the pairs of pronouns and potential antecedents to be classified.

7.4.2 Features

Preiss uses as features the same linguistic properties that Kennedy and Boguraev base their salience hierarchy on. However, she does not specify a ranking of any kind, but leaves it up to the classifier

to include its own ranking in the model it computes. Preiss defines a total of 13 features, out of which one is a unary feature that describes the type of pronoun (non-reflexive vs. reflexive), while the twelve others are binary features that encode properties of a pair of a pronoun and a candidate antecedent. The features are:

1. **Pronoun type** Values: `refl`, `nonrefl` unary
   Encodes the type of pronoun. As already noted in chapter 2.2, personal pronouns and reflexive pronouns are complementarily distributed with respect to the binding domains within which they may be bound by an antecedent.

   In combination with features that encode information about the local syntactic context, this feature should enable the classifier to build a model of relevant binding principles.

2. **Grammatical function** Values: $0 \leq n \leq 31$ binary
   Encodes the distance in sentences between the pronoun to the next antecedent in the coreference chain with the given grammatical function. If both are located in the same sentence, the value is 0. Antecedents further apart than 30 sentences are ignored, and the corresponding feature value is set to 31. One distance feature is extracted for each of the following grammatical functions:

   - subject
   - existential construct
   - possessive
   - direct object
   - indirect object
   - oblique
   - non embedded
   - non adjunct

   Preiss designed this set of features to be accumulative. This means that if a pronoun is member of a *coreference chain*, the grammatical function slots are filled with the distance to the closest element in the chain with the given grammatical function.

The purpose of the grammatical function features is twofold: On the one hand, they provide the classifier with information to determine an internal salience ranking based on grammatical function. On the other hand, the accumulative nature of the features in combination encodes a notion of discourse history, in the sense that entities that are referred to frequently in a discourse are likely to be continued to by referred to, which increases the likelihood of an element of the corresponding coreference chain to become an antecedent.

3. **Parallelism** Values: `parallel`, `nonparallel` binary
   Indicates whether the pronoun and the candidate antecedent have the same grammatical function.

4. **Locality** Values: `nonlocal`, `local` binary
   Encodes whether the pronoun and the antecedent occur in the same local syntactic context. Together with the pronoun type feature, this is one of the features that represent binding properties.

5. **Cataphora** Values: `noncataphoric`, `cataphoric` binary
   Represents the relative location of pronoun and antecedent. Rule-based systems such as RAP or Kennedy and Boguraev's include rules that explicitly penalize cataphoric configurations. Given this feature, the classifier should be able to extract a proper model of the sub-optimality of cataphora, even though this is not explicitly stated in the feature.

6. **Coreference** Values: `yes`, `no` binary
   This feature indicates in the training data whether a pair is coreferent or not.

7.4.3 Evaluation

Preiss (2002) carries out two experiments with different input data. As mentioned before, the parser that she uses attempts to deliver full parsers, but when it fails, it resorts to assigning partial parses to structures it is able to match. In the first experiment, all markables are included in the resolution, regardless whether they originate from partial or full parses. The second experiment only includes full parses (which Preiss calls the "filtered" set). She performs five-fold cross validation on her corpus, yielding results

for five different fragments. Here, only the average performance on all five folds will be given, the reader is referred to Preiss (2002) for full details.

Kennedy & Boguraev	54.8%
Kennedy & Boguraev (filtered)	62.2%
Preiss	53.2%
Preiss (filtered)	61.2%

Table 7.3: Resolution performance of Preiss' reimplementation of Kennedy & Boguraev's knowledge-poor rule-based approach and her memory-based approach.

Table 7.3 shows the performance of Preiss' reimplementation of Kennedy and Boguraev's algorithm and her memory-based approach. Both systems perform nearly the same, with both the unfiltered and the filtered data set as input. Moreover, the transition from the unfiltered to the filtered input data triggers the same increase in performance in both systems - regardless of their very different design. Preiss performed a two-tailed paired *t*-test on the difference in performance between the rule-based and the machine-learning-based approach which confirms that the differences are not significant, both for the unfiltered and filtered versions. This corroborates Preiss' initial hypothesis that machine learning approaches are capable of reaching comparable, or even equal, levels of performance when presented with carefully engineered features.

Chapter 8

A Hybrid Approach to Pronoun Resolution

In the previous two chapters, we discussed approaches to pronoun resolution that employ two fundamentally different architectures. Rule-based approaches rely on a set of manually developed linguistic rules to make their decisions about coreference, whereas machine-learning-based systems autonomously create an internal model from annotated training data that they draw on to select antecedents.

In this chapter, we are going to present a *hybrid* system to pronoun resolution. It combines, as its name suggests, both architectural strategies in one system and we will examine whether such a system can reach or even exceed the performance levels that are achieved by rule-based systems. As explained previously, with systems based on machine learning, it is possible to avoid the time-consuming step of manual rule development. On the other hand, a well-written linguistic rule can operate very accurately, especially if it applies to a clearly defined, limited domain. By combining both strategies, i.e. using rules to perform limited subtasks that require manual control of linguistic decisions, and employing a trained classifier for the bulk of the work, the system can benefit from the strengths of both approaches.

The architecture of our hybrid resolution system is illustrated in figure 8.1. It consists of a chain of three major modules that incrementally operate on the core data set of the system, the set of pairs of pronouns and candidate antecedents:

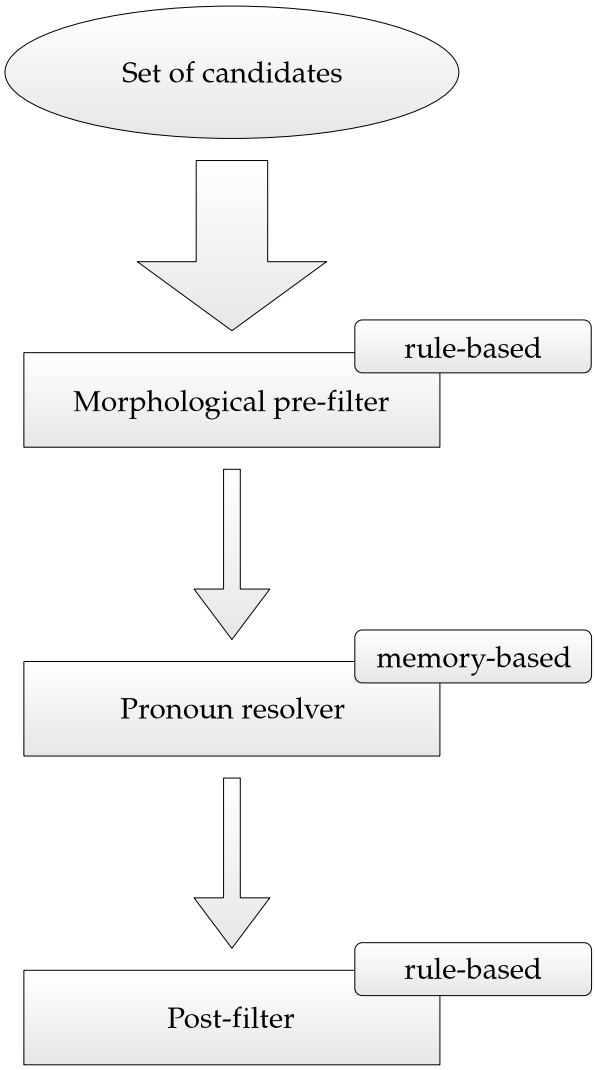


Figure 8.1: Architecture of the hybrid pronoun resolution system

1. The morphological prefilter removes pairs of a pronoun and a potential antecedent (which may either be a definite NP or another pronoun) from the set of candidates if the pronoun and the other NP or pronoun are not morphologically compatible. Since morphological compatibility can be checked with a fairly small set of easily expressible linguistic rules, the implementation of the morphological prefilter adopts a rule-based strategy (see section 8.1).

2. The resolution module proper, which selects antecedents of a pronoun from the set of candidates. This module is based on the TiMBL memory-based learner (see section 8.2). Just as in our rule-based implementation (see section 6.3 in chapter 6), we rely on the gold-standard annotation of expletive pronouns in TüBa-D/Z to exclude them from the resolution process.

3. A postfiltering step that uses heuristics to find an antecedent for pronouns for which the resolution module was not able to find one, and to select one unambiguous antecedent if the pronoun was resolved to more than one antecedents. This module again employs a rule-based strategy (see section 8.3).

There are two factors that are most influential on the output that is generated by the resolver. They are the structure of the training data proper, i.e. the type and the distribution of the samples contained therein on the one hand, and the features that are used to represent the relevant linguistic properties on the other hand. In addition to the description of the design of the resolver component, we will focus on experiments examining the effect of varying these factors on the quality of the resolver output.

8.1 The morphological prefilter

The first module in the processing chain of the hybrid resolver is the morphological prefilter. It removes candidate pairs from the candidate set even before the actual resolution step. It relies on a small set of rules that filter out pairs that are not morphologically compatible. Its purpose is

1. to improve the performance of the overall system by removing pairs that are most likely not coreferent from further processing,

2. to improve the balance between positive and negative training samples in the training data,

3. to reduce the enormous size of the initial candidate set to feasible ranges with respect to both memory requirements and processing time.

We will now turn to each of these tasks in detail.

Removal of unlikely candidates

It is obvious that a given pronoun is *not* in a referential relation with the majority of all other nominal elements in the text. It should be possible for many of these pairs to detect their non-coreference easily with the help of simple surface linguistic properties. Our hypothesis was that especially the morphological compatibility of a pronoun and a candidate antecedent should be a good indicator of their (non-) coreference. Since a pronoun and its antecedent refer to the same extralinguistic entity, they should share the same linguistic properties. A feminine pronoun cannot refer to a masculine antecedent, as this would imply an extralinguistic entity of contradictory gender. Morphological filters of similar kind are employed in many anaphora resolution systems (see chapter 3).

To corroborate the hypothesis, we carried out several experiments that assess the morphological circumstances in our data source, the TüBa-D/Z treebank. We will discuss the results in chapter 8.1.2.

Balancing positive and negative training samples

Machine learning approaches are very sensitive to the distribution of classes in the training data they are presented with. We will discuss this in detail in the section on instance sampling later in this chapter, where we will conduct an in-depth analysis of the influence of the ratio between positive (i.e. the class of anaphoric pairs) and negative (the class of non-anaphoric pairs) samples in the training data on the performance of the memory-based resolver.

A strong predominance of negative samples as it is present in the unfiltered candidate set would lead to an extreme bias of the resolver towards classifying pairs as non-anaphoric. Even though after filtering, the number

Relation	Frequency
anaphoric	13798
cataphoric	1083
bound	904
split_antecedent	48
coreferential[1]	35
instance	10
expletive	1929
none[2]	9141
unannotated	1496
Total	**28 444**

Table 8.1: Distribution of coreference relations with pronouns in the TüBa-D/Z

of negative samples will still exceed the number of positive samples by a factor of four, the classifier that is trained from this filtered set is less biased.

Reduction of candidate set

The initial, unfiltered set of candidates contains 661 205 elements – pairs of all pronouns and all noun phrases within a window of three sentences to the left and to the right of a pronoun. Processing a candidate set of this size requires substantial effort both with respect to processing time as well as memory usage, with the latter being specifically significant in the context of memory-based learning, since all training samples must be stored in memory. The more samples there are, the more memory is needed. Morphological prefiltering helps to remove irrelevant samples from the instance base to minimize memory overhead.

8.1.1 The rule system of the morphological prefilter

The morphological filter makes use of a number of rules that check the morphological agreement of a candidate pair. The rules are hard constraints. This means that a pair that cannot be licensed by any rule will be removed from the set of candidates. We will now discuss the rule system in detail.

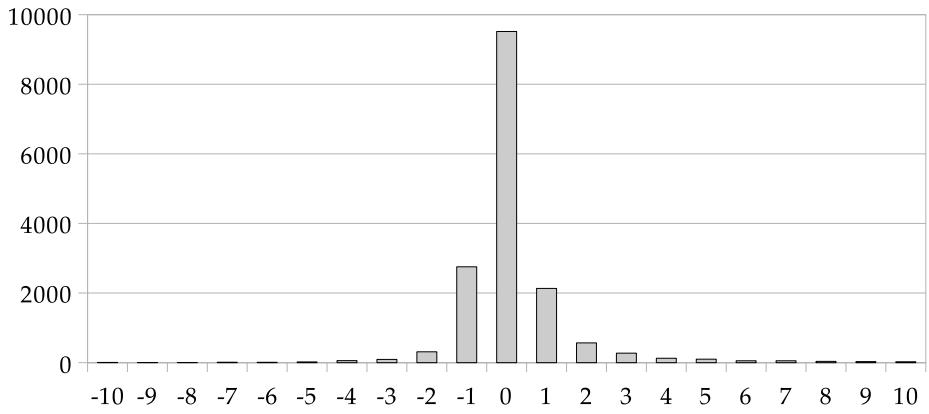


Figure 8.2: Distribution of the distances of the closest morphologically compatible antecedent of a pronoun.

Filter on part of speech of the pronoun

This rule excludes all pairs with types of pronouns other than third person reflexive, personal, or possessive pronouns. Only these types of pronouns are considered in this dissertation.

Filter on distance

The pronoun and the candidate NP must be located within a window of three sentences before or after the pronoun.

We found that the closest morphologically compatible antecedent is located in a window of this size for 97.8% of all pronouns. Figure 8.2 illustrates this distribution. The X-axis plots the distance of the antecedent and the pronoun in sentences. The Y-axis plots the number of antecedents that occur within this distance. It is easily visible that the mass of the antecedents is accumulated within the three sentence window. The majority of all antecedents occurs in the same sentence, and then quickly decays.

Filter on personal pronouns

A pair of a personal pronoun p and a candidate noun phrase n is admitted only if one of the following rules match.[3]

Rule 1
Pronouns
personal pronouns
Candidates
common nouns
proper nouns
personal pronouns
relative pronouns
demonstrative pronouns
POS tags
POS(p) = PPER
POS(n) $\in$ {NN, NE, PPER, PRELS, PDS}
Rule
Number(p) = Number(n) *and* Gender(p) $\equiv$ Gender(n)
or
Number(p) = *plural*

This rule admits a pair of a personal pronoun and a candidate antecedent which is a common noun, a proper noun, another personal pronoun, a (substitutive) relative pronoun, or a (substitutive) demonstrative pronoun if either both elements are singular and their gender feature is compatible, or both elements are plural.

Examples

- ihr/PPER.dsf3 – Mitarbeiterin/NN.gsf *(her – employee)*
  valid: agreement in number *(singular)* and gender *(feminine)*

[3] **A note on notation**: Morphological features may be underspecified. The reflexive pronoun *sich*, for example, can be masculine, feminine, or neuter. To indicate compatibility between two possibly underspecified morphological features f_1 and f_2, we write $f_1 \equiv f_2$. To indicate strong equality, which means that both f_1 and f_2 must have the same value, we write $f_1 = f_2$.

- sie/PPER.nsf3 – die/PRELS.nsf *(she – who)*
  valid: agreement in number *(singular)* and gender *(feminine)*

- sie/PPER.nsf3 – Taake/NE.asm *(she – Taake)*
  not valid: agreement in number *(singular)* but no agreement in gender *(feminine* vs. *masculine)*

- sie/PPER.np*3 – Provisionen/NN.apf *(they – commissions)*
  valid: agreement in number *(plural)*, morphologically compatible in gender (pronoun is underspecified, NP is *feminine)*

Rule 2
Pronouns personal pronouns
Candidates reflexive pronouns
POS tags POS(p) = PPER POS(n) = PRF
Rule Number(p) = Number(n)

This rule admits a pair of a personal pronoun and a reflexive pronoun (as the candidate antecedent) if the two agree in number.

Examples

- ihn/PPER.asm3 – sich/PRF.as*3 *(him – himself)*
  valid: agreement in number *(singular)*

- sie/PPER.np*3 – sich/PRF.as*3 *(they – itself)*
  not valid: disagreement in number (pronoun is *plural*, reflexive is *singular)*

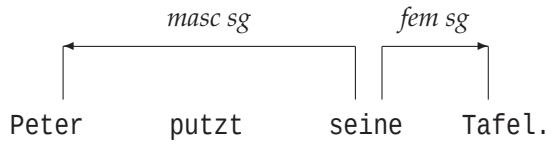
The surface forms of the German reflexive pronoun *sich* are identical in both singular and plural. Since we use manually annotated gold morphology throughout this thesis, we distinguish singular and plural tags.

Personal pronouns and attributive possessive pronouns

Attributive possessive pronouns in German have the property that they agree with both the noun phrase they are an attribute of by means of the morpheme, and with the antecedent by means of the stem:

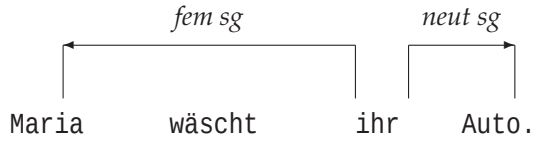
'Peter is washing his car.'

The stem of the possessive pronoun *sein* indicates that the antecedent is either masculine or neuter. The ending, which is a null morpheme in the above example, indicates that the modified noun *Auto* is neuter.



'Peter is cleaning his black board.'

The stem *sein-* again agrees with the antecedent *Peter*. The ending *-e* is the feminine gender marker.



'Maria is washing her car.'

The stem *ihr-* is the feminine stem of the third person possessive pronoun. It agrees with the antecedent *Maria*. The null ending of the possessive pronoun indicates agreement with the neuter noun *Auto*.

The agreement relation of interest for the purposes of the morphological filter is the agreement with the antecedent, which is dependent on the stem of the possessive pronoun. The relevant morphological features are determined as follows:

Stem	Morphology
sein	gender: *masculine*, number: *singular* gender: *neuter*, number: *singular*
ihr	gender: *feminine*, number: *singular* gender: *any*, number: *plural*

Given this, we can now state the compatibility rules for personal pronouns and possessive pronouns.

Rule 3
Pronouns personal pronouns
Candidates possessive pronouns
POS tags POS(p) = PPER POS(n) = PPOSAT
Rule Number(p) = Number(n) *and* Gender(p) $\equiv$ Gender(n)

This rule admits a pair of a personal pronoun and a possessive pronouns as the antecedent if they have the same number and compatible gender.

Examples

- sie/PPER.nsf3 – ihrer/PPOSAT.gpf.sf *(she – her)*
  valid: agreement in number *(singular)* and gender *(feminine)*
- ihn/PPER.asm3 – ihrer/PPOSAT.gpf.sf *(him – her)*
  not valid: agreement in number *(singular)*, but no agreement in gender (pronoun is *masculine*, pronominal NP is *feminine*)

Filter on reflexive pronouns

The filter on reflexive pronouns does not contain any proper morphological rules. There is only one simple syntactic rule that is motivated by a constraint borrowed from binding theory that antecedents of reflexive pronouns must occur in the same clause.

Rule 4
Pronouns reflexive pronouns
Candidates any
POS tags POS(p) = PPER
Rule p and n must be in the same clause

Filter on attributive possessive pronouns

This filter licenses pairs that consist of an attributive possessive pronoun and some other non-pronominal or pronominal noun phrase as a candidate antecedent.

Rule 5
Pronouns attributive possessive pronouns
Candidates common nouns substitutive relative pronouns
POS tags POS(p) = PPOSAT POS(n) $\in$ {NN, PRELS}
Rule Gender(n) = *feminine* *and* Gender(p) = *feminine* *and* Number(p) = *singular* *or* Gender(n) = *masculine* *and* Number(n) = *singular* *and* Gender(p) = *masculine* *and* Number(p) = *singular* *or* Gender(n) = *neuter* *and* Number(n) = *singular* *and* Gender(p) = *neuter* *and* Number(p) = *singular* *or*

Number(n) = *plural* *and* Gender(p) ≡ *any* *and* Number(p) = *plural*

Rule 6
Pronouns attributive possessive pronouns
Candidates proper nouns
POS tags POS(p) = PPOSAT POS(n) = NE
Rule Gender(n) ≡ *feminine* *and* Gender(p) ≡ *feminine* *and* Number(p) ≡ *singular* *or* Gender(n) ≡ *masculine* *and* Number(n) = *singular* *and* Gender(p) ≡ *masculine* *and* Number(p) ≡ *singular* *or* Gender(n) ≡ *neuter* *and* Number(n) = *singular* *and* Gender(p) ≡ *neuter* *and* Number(p) ≡ *singular* *or* Number(n) = *plural* *and* Gender(p) ≡ *any* *and* Number(p) ≡ *plural*

Rule 7
Pronouns
attributive possessive pronouns
Candidates
personal pronouns
substitutive demonstrative pronouns
reflexive pronouns
possessive pronouns
POS tags
POS(p) = PPOSAT
POS(n) $\in$ {PPER, PDS, PRF, PPOSAT}
Rule
Number(p) = Number(n) = *singular* *and*
Gender(p) = Gender(n)
or
Number(p) = Number(n) = *plural*

The plural possessive pronoun is always *ihr* regardless of gender. Therefore, in the plural, it is not necessary to check gender agreement.

We implemented the filter as a Perl program that operates on the analyses of part of speech and morphology in the TüBa-D/Z treebank. In its core functionality, the filter is similar to the work of Filippova (2005), who implements a rule-based morphological filter based on the Xerox Incremental Parsing System (Aït-Mokhtar et al., 2002) as part of her hybrid pronoun resolution system. The rules are formulated as constraints on the morphological features of pronouns and potential antecedents. Filippova uses the gold morphology contained in the TüBa-D/Z treebank as input to the filter component.

Trushkina (2004) presents a hybrid system for morphological analysis and disambiguation. Unlike the author's and Filippova's filters, the system can process raw text. It operates in multiple stages. The first stage is a module for determining *all possible* (i.e. ambiguous) morphological analyses. This module is based on the morphological analysis component that is part of XIP. The second stage resolves ambiguities on the part of speech and morphology levels. This step is rule-based, similar to our implementation. Trushkina then adds a final step of disambiguation which makes used of Probabilistic Context-Free Grammar (PCFG) framework.

Müller (2007) discusses a morphological disambiguation component that is based on a system of cascaded finite state transducers. It takes ambiguous morphological analyses produced by the DMOR morphological analyzer (Schiller, 1995) as its input. It then reduces ambiguity firstly on a chunk-local level by employing a ranking approach that selects from a set of possible morphological tags the one that occurs most often within a chunk (based on the insight that determiners, adjectives and nouns must agree in number, gender, and case within a chunk). Secondly, remaining ambiguities are reduced on the clause level by taking into account the required subject-finite verb agreement.

While the author's and Filippova's systems are specifically designed to act as morphological filters in the context of a pronoun resolution system, the two latter systems can be used as standalone systems that are capable of producing morphological annotations on raw text.

8.1.2 Evaluation of the morphological filter

The aim of the morphological filter is to reduce the size of the set of candidates by removing pairs that are not morphologically compatible. If coreference could be determined only on morphological grounds, the optimal solution would be for the filter to only retain those pairs that are in fact in a relation of anaphoricity or cataphoricity, and remove everything else. This is of course a purely theoretical consideration - if coreference and anaphora resolution only depended on morphology, the problem would be trivial to solve. The result that at best can be achieved in practice is that the majority of correct (i.e. coreferent) pairs are retained in the candidate set, and as many incorrect pairs as possible are removed. However, if in doubt, it is better to keep too many pairs in the candidate set than to remove a correct one with the potential consequence that in the downstream system, a pronoun cannot be resolved at all, because the relevant candidate has erroneously been filtered out at the beginning of the process. So apart from the purely quantitative behavior of the morphological filter, it is of great importance to investigate its qualitative properties. This section will cover both: It will start out with a quantitative evaluation of the filter, inspecting the amount of pairs that are retained or filtered with respect to their morphological compatibility and to their actually being coreferent. The second part of this section will then move on to a qualitative discussion - mainly

focusing on the pairs that are erroneously removed from the candidate set, and discussing the reasons why this happens.

Quantitative evaluation of the morphological filter

In this dissertation the types of pronouns that are considered are restricted to third person reflexive, possessive, and personal pronouns, which amount to a total 13 278 occurrences in the subset of TüBa-D/Z. Thereof, personal pronouns constitute the largest fraction with 7 213 occurrences, followed by possessive pronouns (3 109 occurrences), and reflexive pronouns, which can be found 2 956 times.

For the quantitative evaluation, we computed the set of all possible pairs consisting of each of these 13 278 pronouns and all markable pronominal or non-pronominal noun phrases that are located within a window of three sentences to the left or to the right of the pronoun. This yields a total number of 661 205 pairs. 39 527 of these are in a coreference relation. For this analysis we included all types of referential relations that are annotated in the TüBa-D/Z treebank (for a detailed list of the distribution of the pronouns over the relation types refer to table 8.1 on page 153). The remaining majority of 621 678 pairs is not coreferent. This confirms what has been said on the skewed nature of the data set in the introduction to this section. There is only a small fraction (6%) of positive, i.e. actually coreferent pairs that can be found in the data. The number of negative pairs amounts to 94% of all pairs, that is it exceeds the number of positive pairs by a factor of more than 15.

In addition to counting coreferent and non-coreferent pairs, we assessed the frequency of morphologically compatible and morphologically incompatible pairs, and found that the set of candidates is more evenly distributed with respect to morphological compatibility than with respect to coreference. 302 362 of the 661 205 pairs found in the TüBa-D/Z data are morphologically compatible, which a bit less than half of all pairs. The rest of the pairs are morphologically incompatible.

The two dimensions coreferentiality and morphological compatibility partition the set of pairs in four distinct classes:

1. coreference and morphological compatibility

2. coreference and morphological incompatibility

3. non-coreference and morphological compatibility
4. non-coreference and morphological incompatibility

The pairs that fall into the first class are the pairs that are both coreferent and morphologically compatible. It is for these pairs that the morphological agreement hypothesis of coreference holds. The second class contains pairs that are coreferent, although they are *not* morphologically compatible. In other words, these are the counterexamples for the agreement hypothesis. With respect to the cardinality of these two classes, in order for the morphological filter to work correctly, the number of pairs in the first class should be substantially larger than the number of elements in the second class – as a matter of fact, the cardinality of the second class should be minimal. The number of elements in class 2 is only 1 891, which is a low number. In the following section, we will pay special attention to the pairs in this class and analyze the reasons which lead to occurrences of this combination.

Classes 3 and 4 contain the pairs that are not coreferent. As seen above, the number of elements in classes 3 and 4 exceeds the number of elements in classes 1 and 2 by several orders of magnitude, as the majority of all possible combinations are non-coreferent pairs. Class 3, which contains non-coreferent but morphologically compatible pairs constitutes the class of "false positives". These are the pairs that the morphological filter erroneously admits – due to its extremely limited linguistic knowledge about coreference. Since the total number of morphologically compatible pairs is much larger than the total number of coreferent pairs, we must expect the cardinality of this class to be quite high. Class 4 finally are the cases that are correctly ruled out by the filter since they are neither morphologically compatible nor coreferent. The contingency matrix in table 8.2 lists the size of the four classes.

The morphological filter is quite successful in cutting down on the number of pairs: It removes 358 843 incompatible pairs, which leaves the resulting size of the candidate set at 302 362. This is a reduction by 54 %. As expected for the remaining pairs, the number of morphologically compatible but non-coreferent pairs (264 726 pairs) is by a factor 7 higher than the number of pairs that are both morphologically compatible and coreferent (37 636). Class 2 is the class of morphologically incompatible but nevertheless coreferent pairs. Although the members of this class contradict the

	Morphologically Compatible	**Morphologically Incompatible**	
Coreferent	*Class 1* 37 636	*Class 2* 1 891	39 527
Not coreferent	*Class 3* 264 726	*Class 4* 356 952	621 678
	302 362	358 843	661 205

Table 8.2: Contingency matrix of combinations of pronouns and noun phrases along the dimensions coreferentiality and morphological compatibility. The figures next to or below double lines are the sums of the rows or columns.

Precision	12.4 %
Recall	95.2 %

Table 8.3: Precision and Recall of the morphological filter

morphological agreement hypothesis, it contains 1 891 pairs. They are removed from the set of candidate pairs, and are lost in the subsequent stages of processing as there is no way of recovering them later. In the next section, the qualitative discussion of the morphological filter, we will specifically focus on these cases and track down their origin.

It is further instructive to look at the figures for precision and recall. As shown in table 8.3, the recall is very high (95.2 %) while precision is quite low (12.4 %). For a complete resolution system, this result would be utmost unsatisfactory, but for a filter, this pattern is in fact quite common. The combination of precision and recall can be taken as a measure of the filter's "strictness". High recall but low precision indicates a very conservative setting of the filter: it admits quite a large number of cases that are actually negative examples (therefore the precision is low), where at the same time, the number of pairs that are filtered out even though they should not

have been is kept low (therefore the recall is high). Increasing the filter's strictness increases precision (that is, more incorrect pairs get stuck in the filter). However, this goes along with a decrease of recall, meaning that the fraction of pairs that are removed even though they should not have been removed gets higher.

For the current filter, a conservative setting is superior over a stricter setting. The filter aims to reduce a set of candidate pronoun–antecedent pairs based on the hypothesis that morphological agreement between the pronoun and the NP is a surface hint on their coreference status. As this is quite a weak criterion for the determination of coreference, it is better not to filter too rigidly to make sure that the majority of correct pairs actually remains in the set of candidates.

Qualitative evaluation of the morphological filter

In the previous section, we discussed the quantitative properties of the morphological prefilter on the basis of a number of studies that examine the filtering properties of the system. They show that the prefilter, even though its linguistic knowledge is extremely limited, is quite successful in reducing the size of the candidate set: More than 50 % of the pairs that are not coreferent are removed from the set of candidates, while at the same time, the prefilter retains the majority of those pairs that are in fact coreferent. But even after filtering, the remaining set contains many more pairs that are not coreferent than pairs that are. This reflects the fact that the purely morphological rules that are used by the filter are far too weak to capture the linguistic phenomenon of anaphora: Morphological compatibility is (for most cases) a necessary condition, but not a sufficient one.

As discovered in the previous section, there are 1 891 pairs within the three sentence window that should have remained in the set of candidates because they are coreferent, but were removed from the set by the prefilter because there is no morphological agreement. In the resolution process as a whole, the morphological filter acts like a set of hard constraints that are imposed on the pairs that run through the process. Once removed, there is no way to get them back. Thus, at this point, the morphological filter introduces a system-global source of error. Therefore it is instructive to examine the reasons why the filter is not able to handle these pairs correctly.

A closer inspection of the erroneously filtered pairs shows that the er-

rors can be traced back to the following four systematic classes of problems:

1. morphological incompatibility within coreference chains
2. co-ordinations of singular elements in one antecedent referred to by a plural pronoun
3. errors in the gold standard annotation
4. unsupported types of noun phrases

We will now look at each of these classes in detail.

Morphological incompatibility within coreference chains. The pairs that belong to the first class do not agree morphologically, but are nevertheless coreferent. At first sight they may seem to be counterexamples to the morphological agreement hypothesis. The example below shows the relevant members (printed boldface) of a coreference chain that contains a morphologically incompatible pair:

(1) Im Januar hat **[die Arbeiterwohlfahrt Bremen] [ihren]** langjährigen Geschäftsführer Hans Taake fristlos entlassen. [...] Vorwurf Nummer 1: 165.000 Mark aus der bundesweiten Geldsammlung für die Flutopfer in Südpolen seien über das Konto **[des Bremer Landesverbandes der AWO]** an die Caritas in Danzig geflossen, "damit dort ein Altenheim gebaut wird".

'In Janary the Worker's Welfare Association of Bremen laid off its CEO Hans Taake. ... Accusation Number 1: 165.000 German Marks from the country-wide fund-drive for the victims of the flood in South Poland were paid to the Caritas in Gdansk using an account held by the WWO Bremen "to build a home for the elderly there".'

The morphological features of the three markables are:

- die Arbeiterwohlfahrt Bremen – *feminine singular*
  the Worker's Welfare Association of Bremen
- ihren – *feminine singular*
  its
- des Bremer Landesverbandes der AWO – *masculine singular*
  the Bremen branch of the WWO

The two definite noun phrases *die Arbeiterwohlfahrt Bremen* and *des Bremer Landesverbandes der AWO* refer to the same entity and are therefore coreferent. While the former noun phrase has feminine gender, the latter NP is masculine. This is not uncommon for coreferent definite noun phrases; in fact it is more the rule than the exception. In other words, the morphological agreement hypothesis does not hold for coreferent definite NPs.

Since we assume a binary classification model for determining coreference between pronouns and antecedents, we create one tuple for any one pronoun and any possible noun phrase:

1. die Arbeiterwohlfahrt Bremen/*fem sg* – ihren/*fem sg*

2. des Bremer Landesverbandes der AWO/*masc sg* – ihren/*fem sg*

The NPs *die Arbeiterwohlfahrt Bremen* and *des Bremer Landesverbandes der AWO* are coreferent, thus they belong to the same coreference set. The pronoun *ihren* is a member of this same coreference set as well. In our binary representation a pair is a positive pair if the pronoun and the antecedent belong to the same coreference set. This is the case, in both examples above. Thus, both pairs are positive pairs.

However, the morphological filter removes the second pair, because the pronoun and the candidate do not agree. From the point of view of morphological compatibility, this is the correct decision. Nevertheless, in our evaluation, we count this as a recall error, because what we evaluate is the membership in a coreference class, as annotated in the gold standard. What we see here is a target conflict between discovering as many elements of a coreference set as possible given the binary representation, and determining the subset of those elements that additionally are morphologically compatible.

Co-ordinations of singular NPs in one antecedent referred to a plural pronoun. Co-ordinated noun phrases are frequently used to express reference to a whole set of entities, such as in the following example:

(2) Da treten nämlich [$_{NX}$ [$_{NX\text{-}KONJ}$ der Bariton Renato Mismetti] und [$_{NX\text{-}KONJ}$ der Pianist Max Daniel]] auf.

'The baritone Renato Mismetti and the pianist Max Daniel perform there.'

The coordinated NP *der Bariton Renato Mismetti und der Pianist Max Daniel* refers to the set of entities that consist of the two individuals *Renato Mismetti* and *Max Daniel*.

In the sentence following (2), the plural possessive pronoun *ihren* is anaphoric to the complete noun phrase *der Bariton Renato Mismetti und der Pianist Max Daniel*:[4]

(3) Beide haben sich gründlich mit der Musik [ihres] Landes beschäftigt.

'Both have studied thoroughly the music of their country.'

In the co-ordinated NP, there is no plural head that would enable the morphological filter to license the pair.

Errors in the gold standard annotation. Morphological mismatches that occur in the third class must be attributed to errors in the manual gold standard annotation – both on the morphological layer and the coreference layer. In example (4), the personal pronoun *sie* is anaphoric to *die Verwaltungsangestellte*.

(4) "Betroffen ist man", sagt [die Verwaltungsangestellte/n s f] trocken, manches rühre [sie/a p *3] auch nach zehn Jahren Dienst noch.

' "It strikes you", says the employee of the administration - that she is still moved sometimes, even after ten years in office.'

The feminine singular and plural surface forms of the personal pronoun *sie* are the same, and the annotators mistook plural for singular with the pronoun.

Unsupported types of antecedents We excluded some types of noun phrases from being considered as antecedents in this thesis which are:

- PWS - substitutive wh-pronouns
- FM - foreign language material
- CARD - cardinal numbers

[4] It is also anaphoric to the pronoun *Beide*. However, since this does not pertain to the present problem, we omit the markable in this discussion.

8.2 The memory-based resolution module

The second component in the processing chain is the module that performs the actual pronoun resolution. It is based on memory-based learning (Daelemans et al., 2005), the concepts of which were introduced in the previous chapter.

The memory-based resolution module which we are going to discuss in this section assumes a processing model that is fundamentally different from that of the rule-based approach presented earlier. The rule-based system aims to solve the resolution problem by taking into account the salience of multiple competing referents, which not only is determined by the static morphosyntactic structure of the text but also by the dynamically changing discourse model: The salience of a referent increases with the number of times it is referred to in the discourse. To summarize, the rule-based approach selects an antecedent from a set of competing antecedents. One might say that the rule-based approach makes its decisions based on a global view of the discourse.

The existence or non-existence of a relation of anaphora or cataphora between a pronoun and a noun phrase partitions the input data into two disjoint classes. The task of pronoun resolution can thus be reformulated as a binary classification problem of assigning each pair one of two possible classes: The positive class (which represents coreference), and the negative class (which represents non-coreference). This is a vital prerequisite for the employment of a machine learning approach such as memory-based learning whose core mode of operation is to assign the samples it is presented a class from a predefined space of classes. The memory-based resolution module adopts a pairwise resolution model of the kind outlined in chapter 3.

8.2.1 Input data

The raw input data for the resolution module are the pairs of pronouns and candidate antecedents which have been filtered by the morphological prefilter. For each pair, a configurable set of features is extracted and then bundled in a feature vector. The resulting feature vectors, one vector for each pair, serve as the input to the TiMBL memory-based learner. The feature vectors are represented in a text based format, one vector per line, with

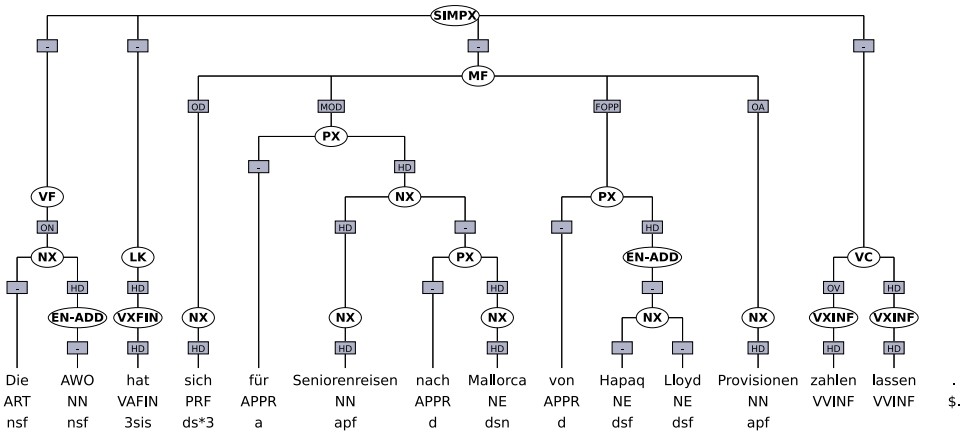


Figure 8.3: Parse tree of the sentence in example (5).

```
0001,119,118,refl,ana,diff,loc,-2,ON,OD,proper,def,top,yes
0001,119,120,refl,cata,diff,loc,-2,HD,OD,common,na,top,no
```

Figure 8.4: Feature vectors for the positive pair *sich – die AWO* and the negative pair *sich – Seniorenreisen*.

features separated by commas.

Figure 8.4 shows two feature vectors that correspond to two candidate pairs in the following text fragment, whose syntactic annotation is shown in figure 8.3.

(5) **[Die AWO]**$_i$ hat **[sich]**$_i$ für **[Seniorenreisen nach Mallorca]**$_j$ von Hapaq Lloyd Provisionen zahlen lassen.

'The WWO accepted commisions from Hapag Lloyd for senior's trips.'

In the example, the reflexive pronoun *sich* is anaphoric to *die AWO*, which yields a positive pair. The pronoun is not in a referential relation to *Seniorenreisen* (the candidate relation here would be *cataphoric*), so this gives a negative pair.

The first feature vector in figure 8.4 corresponds to the positive pair *sich – die AWO*. This is indicated by the `yes` class label, which is the last feature in the vector. The second feature vector encodes the negative pair *sich – Seniorenreisen*. The class label here is `no`.

The other features encode properties of the pronoun and the noun

phrase that are members of the pair. The first three features in the vector are for information purposes only – the TiMBL software is directed to ignore them.

The first feature in the feature vector is the number of the article that the pair occurs in.

The second and third features are the markable IDs of the pronoun and the candidate antecedent.[5]

Features 4 and 5 represent properties of the pronoun. Feature 4 encodes the direction of the potential relationship by the relative position of the pronoun and the candidate antecedent. In the first feature vector, the value of this feature is `ana` because the antecedent *die AWO* occurs to the left of the pronoun. In the second feature vector, the feature value is `cata` because the (negative) instance would be a cataphoric configuration, since the NP *Seniorenreisen* follows the pronoun.

The next three features represent properties of the pair. The value `diff` means that the grammatical functions of the pronoun and the potential antecedent are different. Since the candidate NP and the pronoun both occur in the same local clause, the value of feature 7 is `loc`. Feature 8 encodes the distance between the pronoun and the candidate antecedent in words.

The values of features 9 and 10 are the grammatical functions of the candidate antecedent and the pronoun, respectively. In our example, the noun phrase *die AWO* is the subject of the sentence. Subjects are marked with the function `ON` in TüBa-D/Z. The pronoun *sich* is dative (label OD). As shown in figure 8.3, the NP *Seniorenreisen* is the head of the noun phrase *Seniorenreisen nach Mallorca*, therefore the value for the its grammatical function feature is `HD`.

The three features 11 to 13 are properties of the noun phrase. Feature 11 encodes whether the noun phrase is a proper noun (feature value `proper`), or a common noun (value `common`). Feature 12 describes the definiteness of the article that occurs together with the NP, if any. The NP *die AWO* comes with a definite article, therefore the value of the feature is `def`. The NP *Seniorenreisen* does not have an article at all, therefore the value of the feature is `na`. The value `top` of feature 13 finally indicates that the NPs are not embedded in other NPs.[6]

[5] Each markable in the corpus is assigned a unique numeric ID.

[6] By "embedding" we mean that there is no other intervening NX node on the path from the NP to the closest dominating field node.

8.2.2 Baseline

For evaluation, we split the data set as described in ten parts of equal size for 10-fold cross validation. Since classification takes place only pair-wise, it was not necessary to pay attention to article boundaries. Each training set contains 90% of the total number of pairs, which amounts to 1 461 530 training instances. The remaining 10% are assigned to the test sets, which then contain 162 390 pairs.

In order to be able to judge the performance of the memory-based resolver, we computed two baselines. For each of the ten test sets, pronouns were resolved either (a) to the **closest noun phrase** or (b) to the **closest subject**. We determined the two baselines using a simple Perl program that accepts pronoun-candidate pairs in the same format as the TiMBL classifier and "classifies" only those pairs as positive pairs that involved either the closest NP or the closest subject. We evaluated the output with the same program as we evaluated the TiMBL output.

Note that since the pairs in the training and test sets already passed the morphological prefilter, they are all morphologically compatible.

Apart from morphological compatibility, the closest NP baseline does not take into account any more higher-level linguistic information. Therefore, we expect this baseline to constitute a lower bound for the performance of the resolution system. Subjects on the other hand play an important role in the linguistic description of anaphora. The referents that are realized by noun phrases in the subject position are usually highly salient in a discourse. Therefore, they are frequently referred to by pronominalized expressions. Therefore we expect the closest subject baseline to be considerably stronger than the closest NP baseline.

The results of the baseline experiments are summarized in table 8.4. Generally, the results are as expected. First of all, both baseline resolvers make a positive decision only for a very small number of all possible pairs. The closest NP resolver classifies only 842 potential pairs as anaphoric, and the closest subject resolver makes a positive decision for only 835 pairs. For both baseline approaches, this is only a fraction of all potential pairs of 0.5%. This is directly related to the weak resolution strategy employed for both baselines, which select exactly one antecedent for each pronoun and do not try to find more antecedents that are further apart.[7] There are

[7] We adopt a purely pairwise resolution strategy that does not attempt to reconstruct

Closest NP Baseline	
Correctly classified as anaphoric	499
Incorrectly classified as anaphoric	343
Total classifications	842
Precision	0.593
Recall	0.016
F-Measure	0.032
Closest Subject Baseline	
Correctly classified as anaphoric	668
Incorrectly classified as anaphoric	167
Total classifications	835
Precision	0.800
Recall	0.022
F-Measure	0.042
Total number of coreferent pairs	30721
Total number of pairs	162390

Table 8.4: Baseline results

842 distinct pronouns in the test data, which is the maximum number of pairs that can be found. The closest subject approach only finds 835 positive instances. For the remaining 7 pronouns, there is no morphologically compatible subject within the sentence window, which means that these pronouns cannot be resolved at all.

The small number of pairs that are resolved results in a low recall of 0.016 for the closest NP baseline, and a slightly higher recall of 0.022 for the closest subject baseline. Even though the closest subject baseline resolves fewer pronouns that the closest NP baseline, the recall of the former is higher. This is because in discourse, highly salient referents frequently tend to be realized in subject position. The heuristic of picking the closest subject therefore approximates anaphora better than just selecting the arbitrary NP that is closest to the pronoun.

This insight is corroborated by comparing the precision of the closest

coreference chain information. However, since in the data, an instance is created for each possible pair, coreference chain information is implicitly represented by the union of all noun phrases that a pronoun is resolved to.

subject baseline with the precision of the closest NP baseline. The former reaches a value of 0.8, which is substantially higher than the precision of the closest NP baseline (0.593). While the closest NP approach only classifies roughly half of the 842 pronouns correctly, the closest subject approach is correct in 80%.

Note that the very low recall is also due to the heuristic used in the baseline approach: *Only one* antecedent is selected for a pronoun. All other markables, among which there might be additional antecedents (but which are not the closest subject or the closest NP) are ignored. So, even though the heuristic is fairly accurate in the few NPs it actually selects, it is too weak to discover potential antecedents in locations other and more complex than closest subject or closest NP. Hinrichs et al. (2005a) report a baseline that picks the closest subject as well. This baseline reaches a precision of only 0.5, but a substantially higher recall of 0.647, resulting in an f-measure of 0.564. This difference is to be attributed to a different strategy of setting up the test data. While Hinrichs et al. (2005a) filter their test set such that from a set of markables that share the same head only the one spanning the largest amount of words remains in the test set, we leave all markables in the set, thus creating a larger number of positive instances. In our approach more positive pairs with antecedents in positions other than ON are included in the test data which can not be discovered using our closest subject baseline.

To summarize, to determine the lower bound of the pronoun resolution we consider two baselines of different strength. The closest NP baseline does not use any linguistic information relevant for anaphora (apart from morphological compatibility), and is therefore the lowest possible bound. The closest subject heuristics employed to determine the second baseline is stronger due to the usually high salience of referents that are realized in the subject position. Due to the way they are designed, both approaches fail to discover more than one antecedent even though there might be more referents that belong to the same chain.

8.2.3 Feature set

In this section, we will discuss the features that are presented to the TiMBL learner individually and list all possible values they can take.

The first three features ARTNUM, PRONID, and NPID are only used for

Feature	Description
ARTNUM	article number
PRONID	pronoun markable ID
NPID	NP markable ID
PRONTYPE	pronoun type
PRONGF	grammatical function of the pronoun
NPGF	grammatical function of the noun phrase
NPTYPE	type of NP
DEFINITENESS	type of article
EMBEDDING	embedding of NP
DIRECTION	direction of relation
PARAGF	parallelism of grammatical function
SENTDIST	sentence distance
WORDDIST	word distance
GOLDCLASS	gold referential relation

Table 8.5: Features used for the memory-based resolver

internal bookkeeping and TiMBL is instructed to ignore them.

Feature ARTNUM

ARTNUM	**article number** (ignored)
Values	$1 \ldots n$
	The number of the article that the pair occurs in.

Feature PRONID

PRONID	**pronoun markable ID** (ignored)
Values	$1 \ldots n$
	The markable ID of the pronoun in the current article.

Feature NPID

NPID	**NP markable ID** (ignored)
Values	$1 \ldots n$
	The markable ID of the noun phrase in the current article.

Feature PRONTYPE

PRONTYPE	**pronoun type**
Values	`refl pers poss`
	The type of pronoun.

The pronoun type is determined by the STTS POS label of the pronoun, as annotated in the gold morphology of the treebank. The mapping is as follows:

STTS tag	**Pronoun type**
`PPER`	`pers`
`PRF`	`refl`
`PPOSAT`	`poss`

Feature DIRECTION

DIRECTION	**direction of relation**
Values	`ana cata`
	The direction of the relation, starting at the pronoun, pointing towards the NP

The direction feature encodes the direction of a potential referential relation. It is determined by the relative position of the pronoun and the candidate NP. If the NP is located to the left of the pronoun, the value is set to `ana` (for anaphoric). If the NP is located to the right of the pronoun, the feature takes the value `cata` (for cataphoric).

Feature PARAGF

PARAGF	**parallelism of grammatical function**
Values	`para diff`
	Equality of grammatical function

If the grammatical functions of the pronoun and the candidate NP are equal, this feature takes the value `para`. If the grammatical functions are different, the value is `diff`.

Feature SENTDIST

SENTDIST	**sentence distance**
Values	`loc` 0...n
	The distance in sentences between the pronoun and the noun phrase

The sentence distance between the pronoun and the candidate antecedent. The value `loc` indicates that the pronoun and the potential antecedent are in the same local clause. If the pronoun and the potential antecedent occur in the same sentence (but not in the same clause), the value is 0. Higher values indicate larger distances. The value of this feature is always positive regardless of the relative positions of the pronoun and the noun phrase.

Feature WORDDIST

WORDDIST	**word distance**
Values	0...n
	The distance in words between the pronoun and the noun phrase

The word distance between the pronoun and the candidate antecedent. Just like the SENTDIST feature, the sign of this feature is not affected by the relative position of the pronoun and the NP, and remains always positive.

Feature PRONGF

PRONGF	**grammatical function of the pronoun**
Values	`- -- APP FOPPK HD KONJ MOD MODK MOD-MOD OA OADJP-MO OAK OA-MOD OD ODK OD-MOD OG OG-MOD ON ONK ON-MOD OPP-MOD OS OS-MOD PRED PREDK PRED-MOD V-MOD`
	The grammatical function of the pronoun

Feature NPGF

NPGF	**grammatical function of the noun phrase**
Values	`- APP HD KONJ OA OD ON ON-MOD PRED PRED-MOD V-MOD`
	The grammatical function of the noun phrase

Feature NPTYPE

NPTYPE	**type of NP**
Values	`common proper`
	The type of the candidate antecedent: proper or common noun.

Feature DEFINITENESS

DEFINITENESS	**type of article**
Values	`def indef na`
	Definiteness of the noun phrase

This feature encodes whether the candidate antecedent has a definite article (`def`), an indefinite article (`indef`), or no article at all (in this case, the value of the feature is `na`).

Feature EMBEDDING

EMBEDDING	**Embedding of NP**
Values	`top embedded`
	Encodes whether the noun phrase is embedded in another noun phrase or not.

Feature GOLDCLASS

GOLDCLASS	**gold referential relation**
Values	`yes no`
	The gold annotation whether the pronoun and the NP are in fact anaphoric.

This feature is extracted from the manual coreference annotation in the corpus. It takes the value `yes` if the noun phrase is in fact an antecedent of the pronoun, and `no` if it is not. This feature is only used when training the machine learner and for evaluation, but ignored during the testing phase.

8.2.4 Experiments and evaluation

The actual evaluation was carried out on the same data set that was also used to determine the baselines. To reiterate, it consists of 174976 pairs that were extracted from the manually annotated TüBa-D/Z. For 10-fold cross validation, ten partitions of the data were created, with the training partition containing 90%, and the test partition containing the remaining 10% of the pairs.

For each fold, the TiMBL memory-based learner (Daelemans et al., 2005) is used to classify the samples in the test fraction, after having been trained on the corresponding training section. TiMBL assigns each pair one of the class labels `yes` or `no`. The label `yes` indicates that TiMBL classifies a pair as anaphoric or cataphoric.[8] The negative class of pairs that are not considered anaphoric is represented by the label `no`.

The TiMBL software offers a wide range of settings that influence the classification process, which we explained in chapter 7. For the present experiments, the settings were chosen as follows:

- **Classification algorithm**

  For the classification algorithm, the IB1 algorithm was chosen, which is a modified variant of the standard k nearest neighbors algorithm for instance based classification. IB1 differs from the standard k nearest neighbors algorithm in that it does not consider the k nearest neighbors proper, but rather all neighbors with k closest distances to the sample to be classified (see chapter 7).

  For the current experiments, a value of $k = 20$ proved empirically optimal.

- **Distance metric**

  The distance metric for measuring the distance of two instances in the feature vector space was set to *modified value distance metric*.

- **Feature weighting**

  For the computation of the distance between two feature vectors, TiMBL introduces a weight of importance on each feature that scales

[8] The actual type of relation is implicit in the relative location of the pronoun and the antecedent.

	Prec.	**Rec.**	**F**	**# resolved**
Closest NP	0.569	0.016	0.031	852
Closest subject	**0.773**	0.022	0.042	846
TiMBL resolver	0.664	**0.428**	**0.521**	**19391**

Table 8.6: Performance of the TiMBL-based resolution module.

the distance between two features. Features that are more important receive a higher weight than less important features which results in a greater total distance between two features vectors when important features differ.

The distance metric we use here is *Gain Ratio*, a version of the *Information Gain* metric that looks at the influence of an individual feature on the overall class of a sample, normalized with respect to the number of different values that a feature can take (see chapter 7).

All features shown in table 8.5 were included in the experiment, and TiMBL was run on each of the ten test partitions in the 10-fold data set.

For all samples in all the folds, we computed two total values of precision (P) and recall (R) according to the following formulae:

$$P = \frac{\sum_k TP_k}{\sum_k (TP_k + FP_k)}$$

$$R = \frac{\sum_k TP_k}{\sum_k (TP_k + FN_k)}$$

where k is the k^{th} fold, TP_k the number of pairs in the k^{th} fold that were correctly classified anaphoric (true positives), and FP_k the number of pairs in the k^{th} fold that were incorrectly classified anaphoric (false positives). FN_k is the number of pairs in the k^{th} fold that were not classified anaphoric by TiMBL, but that actually are.

The result of the experiments is shown in table 8.6, together with the two baselines closest NP and closest subject.

If one compares the results of the TiMBL resolver with the two baselines, the most notable difference is the recall, which increases substantially. While the two baseline approaches only classified between 1% and 2% of all actually anaphoric cases, the memory-based resolution module finds

almost 43%. This must be attributed to the fact that the memory-based resolver is not stuck with mechanically selecting the closest noun phrase or subject as the antecedent, but it can capture the more complex instances of anaphoric pairs. Furthermore, the resolution model of the memory-based resolver differs from that of the baseline approaches. The latter only select exactly one antecedent for a pronoun and then stop searching for more possible antecedents, while the memory-based resolver can find more than one antecedent.

The precision that is achieved by the memory-based resolver lies between that of the closest NP baseline and the precision of the closest subject baseline. Compared to the closest subject baseline, precision drops by more than ten points of percentage.

To determine the relative importance of the features, we repeated the experiments as described above, each time leaving out one feature, and recorded the modified performance of the resolver module. This strategy is called *reverse feature selection*. The principle is to start out with the full set of features, which yields the optimal results. Leaving out one feature at a time decreases the performance by a certain amount. The higher this amount, the more important the information that the feature contributes. Table 8.7 summarizes the result of the feature selection.

	Prec.	**Rec.**	**F**
TiMBL resolver	0.664	0.428	0.521
Missing feature			
DEFINITENESS	0.689	0.345	0.459
DIRECTION	0.661	0.413	0.509
EMBEDDING	0.664	0.414	0.510
NPGF	**0.626**	**0.219**	**0.324**
NPTYPE	0.664	0.420	0.514
PARAGF	0.664	0.428	0.521
PRONGF	0.667	0.425	0.520
PRONTYPE	0.659	0.410	0.506
SENTDIST	0.663	0.411	0.507
WORDDIST	0.610	0.433	0.506

Table 8.7: Results of the feature selection experiments.

Overall, the performance of the system remains fairly constant regardless whether a feature is missing or present. There is one notable exception to this pattern. When the NPGF feature, which represents the grammatical function of the NP that is the candidate antecedent, is excluded from the classification process, the performance of the resolver breaks down in f-measure from 0.521 to 0.324. Thus substantial loss is mainly due to a considerable decrease of recall from 0.428 down to 0.219. Precision drops as well from 0.664 down to 0.626.

Essentially, this feature corresponds to the concept of a salience hierarchy based on the grammatical function of candidate NPs in rule-based approaches such as Lappin and Leass' presented earlier. The importance of the concept of salience, which has been introduced in rule-based approaches on the basis of linguistic considerations is thus corroborated by the model autonomously acquired by the machine learning system.

8.3 The postfilter

As mentioned before, the memory-based resolver adopts a pairwise model of anaphora resolution. One main characteristic of this resolution model is that the resolver cannot not take into account any context apart from what is present in the feature vector that describes a pair. Specifically, the resolver does not have any information about whether a pronoun has already been resolved and whether an antecedent has been found for a pronoun at all. The purpose of the postfilter is to re-consider some context information to enhance the performance of the overall resolution process by optimizing some of the classifications made by the resolver module.

Resolving a pronoun to more than one antecedent has both advantages and disadvantages. Usually, both coreference and anaphora are considered relations that hold between more than two entities in a discourse, thus forming the coreference sets or coreference chains referred to on multiple occasions in this thesis. Therefore, if a pronoun is resolved to multiple antecedents, it is possible to reconstruct some of the coreference chain information that a resolver that is based on a pairwise resolution model would otherwise not be able to produce.

While it is certainly true for coreference relations of definite noun phrases that their arity is usually greater than two, the nature of anaphora

is actually more that of a binary relation: a pronoun is always interpreted with respect to exactly one antecedent rather than a chain or set. This is corroborated by a finding about the morphological compatibility of a pronoun and its antecedent that was pointed out earlier in this chapter: If every noun phrase that is a member of a certain coreference set is paired with a pronoun that is a member of the same coreference set as well, then some of the resulting pairs might not agree with respect to their morphology. The pronoun cannot be interpreted relative to these noun phrases, even though they do belong to the same coreference class. Moreover, with an increasing number of antecedents that are suggested by the resolver, the chance increases as well that incorrect antecedents are among those suggested. From this point of view, both the necessity and the utility of resolving a pronoun to more than one antecedent become questionable.

The postfilter is applied to the output of the TiMBL resolver and addresses both the issue of pronouns for which no antecedent could be found at all and the the issue of multiple antecedents for a pronoun with a set of simple heuristic strategies as follows.

8.3.1 Unresolved pronouns

If the resolver does not classify any of the potential pairs that are extracted from the text for a given pronoun as anaphoric, then this pronoun remains unresolved. In such cases, the postfilter searches the set of potential candidates for the closest morphologically compatible subject and selects this as the antecedent of the pronoun. The baseline experiments have shown that with the simple approach of always selecting the closest subject as the antecedent, one can reach precision as high as 0.773. The biggest shortcoming of the baseline approach is its low recall. This will not be much of an issue here, since when using this heuristic as a postfilter, in the worst case, the recall remains unchanged (no antecedent found neither by the resolver nor by the postfilter).

8.3.2 Multiple antecedents

If multiple antecedents have been suggested for a pronoun by the resolver module, the *leftmost* antecedent from those suggested is retained as the only antecedent. Schiehlen (2004) reports that for personal pronouns, selecting

	Prec.	**Rec.**	**F**	**# resolved**
TiMBL resolver	0.664	0.428	0.521	19391
Postfilter	**0.741**	**0.930**	**0.825**	9513

Table 8.8: Performance of the resolution system after applying the postfilter

the leftmost antecedent from a list of multiple possible antecedents yields superior results than picking an antecedent in right-to-left order. We empirically found the same results.

8.3.3 Results

Table 8.8 summarizes the results after applying the postfilter. Precision is increased by more than 7 points of percentage. Recall more than doubles. This has to be attributed to two reasons. Firstly, the postfilter finds antecedents for pairs for which the resolver module failed altogether. But secondly, the postfilter *removes* superfluous pairs from the resolved files. For pronouns that were not resolved at all, only one pair is retained by the postfilter - the one with the closest subject. Some of the removed pairs may have been pairs that were actually coreferent, thus counting into the computation of recall. Removing these pairs results in a smaller overall number of pairs, therefore the proportion of resolved pairs compared to the pairs that should be resolved becomes larger.

By removing these, recall is increased. The same happens with the disambiguation of multiple antecedents. Only one pair is retained, while the others are removed.

8.4 Instance sampling

It is a typical property of training data that the training instances are not evenly distributed over the classes that are contained in the data. Usually, there are some classes that many training instances belong to, and others which have very few elements. In our setting there are only two different classes: The class of pairs of a pronoun and an antecedent that *are* in fact coreferent, and the class of pairs that *are not*. Considering the number of entities that a pronoun could possibly refer to in a text, it is not surprising

that in the training data, the number of pairs that are *not* coreferent *(negative instances)* substantially outweighs the number of pairs that *are (positive instances)* – simply because most entities in the text are *not* coreferent with the pronoun. The master instance set that we derive our actual ten-fold training and test sets from contains 30 721 positive (coreferent) samples, and 131 671 negative (not coreferent) samples, i.e. the number of negative pairs exceeds the number of positive pairs by a factor of more than four.

The instance base that is constructed by the TiMBL classifier during the training phase directly reflects this ratio of positive and negative training instances (see section 7.3). This leads to a bias of the system of assigning the negative class to new instances – simply because for each new pair, the classifier finds many more negative similar samples in the instance base than positive ones. Only when the similarity to a positive sample is compelling, TiMBL will classify the new pair as coreferent. From a slightly more abstract perspective, the distribution of the training data as described leads to a very conservative classifier that assigns the positive class only given strong evidence, while otherwise preferring the negative class. This means that many coreferent pairs that do not exhibit very strong features of coreference will not be recognized by the classifier as being coreferent, while on the other hand, for a pair that *is* classified as coreferent it is fairly likely that this decision is in fact correct. In other words, the classifier trades recall for precision.

The results of our experiments show exactly this property: As shown in table 8.11, precision is 66.4%, while recall is much lower at 45.7%. This means that the resolution system works fairly accurate (as measured by precision) – provided that it attempts to resolve a pair at all, which only happens for less than half of the pairs that should be resolved (as shown by the rather low recall). This is to be attributed to the bias of the classifier towards negative classification, which in turn stems from the skewed distribution of positive and negative instances in the training data. A more balanced distribution of training samples would lead to a classifier that is less biased.

A strategy that leads to more balanced training sets is *instance sampling*. It works by removing instances from the training data that belong to high-frequency classes to increase the weight of classes that contain less instances. Zhao and Ng (2007) report that they successfully employed

	Prec	Rec	F
Heuristic baseline	15.0	99.7	26.1
Classifier baseline (r = 1:29.4)	51.1	19.8	28.6
Classifier with Sampling (r = 1:8)	44.3	59.8	50.9

Table 8.9: Performance of Zhao and Ng's resolver for Chinese zero pronouns. The ratio r is the ratio of positive versus negative training samples in the training data.

instance sampling in their system for resolving Chinese zero pronouns. Zhao and Ng employed a machine learning approach that is based on the J48 decision tree algorithm. In the training data they used, there were 343 positive training samples for zero pronouns as opposed to 10 098 negative samples, which is a ratio of 1:29.4. In this configuration, their classifier performed at an f-measure of only 28.6 (see table 8.9), which is only slightly above their baseline that used a very simple rule-based heuristic. For the experiments, Zhao and Ng created training sets of differently strong skewedness, by *randomly* removing negative samples from the original training data. This way, they produced 29 differently balanced training set with ratios between 1:1 and 1:29. They examined the influence of the skewedness of the training data on the performance of their classifier and found that the ratio of positive versus negative samples has substantial impact. The classifier achieved maximum performance at a ratio of 1:8. Here, the system reached an f-measure of 50.9, which is almost twice as high as the baseline performance with ratio 1:29.4, caused by the dramatic increase of recall from 19.8 % to 59.8 %. Precision drops as expected due to the reduced conservativeness, however this is a minor drop compared to the strong increase of recall.

Given Zhao and Ng's encouraging results, we examined the influence of instance sampling on our system as well. We experimented with four sampling methods that differ in what information they rely on in the sampling process. All four methods are instances of *undersampling*, i.e. methods for balancing the training set by *removing negative instances* (Japkowicz and Stephen, 2002). An alternative to undersampling is *oversampling*. Here, training sets are balanced by *re-adding positive instances* to a training set. Müller (2008) successfully employs this technique in his

decision-tree-based system for resolving *it*, *this*, and *that* in spoken dialog.

We will now turn to the detailed discussion of the four variants of instance sampling that we employed.

Proximity sampling. This variant of instance sampling is based on the intuition that anaphoric relations are closely tied to proximity: On the one hand, two entities are more likely to share an anaphoric relation if they are closer, but on the other hand, negative samples in the same region are especially informative on what configuration leads to no relation, in spite of proximity.

Vector-distance sampling. This method takes into account the distance between the feature vectors that represent positive examples and the feature vectors that represent negative examples in the sample space of the memory-based learner. The usefulness of negative examples is determined by their proximity to the positive examples. The intuition here is that a negative example is especially useful if it is very close to positive examples. In terms of the search space, we are trying to concentrate on negative examples that are on the border to the positive class and remove the negative examples that are further apart. We name this method *close distance sampling*.

In order to determine the discriminatory power of negative examples that are further away, we repeated the sampling and chose all the negative examples that were further apart. We will refer to this method as *far distance sampling*.

Incremental learning, IB2. IB2 is a modification of the standard memory-based learning algorithm suggested by Aha et al. (1991), in which the examples are presented incrementally, and only those examples are kept for the training set that are misclassified by the current training set. In our case, we use a slight modification of the algorithm, in which we keep all the positive examples and add the negative ones incrementally.

Random sampling. Random sampling is a method which randomly removes negative examples from the training set until the pre-determined

target ratio is reached. This is the method used successfully by Zhao and Ng Zhao and Ng (2007) as described above.

In the following, we describe the four sampling methods in detail, and then move on to a joint discussion of the results of the experiments that we carried out with each of the four methods.

8.4.1 Proximity sampling

In order to generate the baseline training data, we paired a pronoun with every morphologically compatible NP within the extraction window. The advantage of this approach is that a maximum of information is presented to the TiMBL learner. The disadvantage is that the overweight of negative samples leads to a bias towards negative classification, as detailed in the previous section. Other approaches of machine-learning-based anaphora resolution, such as the one by Soon et al. (2001), choose a different strategy: Given a pair of a pronoun and a correct antecedent, they include as negative samples in the training data only those pairs that are located *between* the pronoun and the correct antecedent and that are not coreferent. This way, the negative overweight in the training data can be reduced. Proximity sampling is a variant of instance sampling that is based on the linguistic insight that anaphora is a phenomenon that is closely tied to proximity. On the one hand, two entities are more likely to be anaphoric if they are closer, but on the other hand, negative samples in the same region are especially informative on what configuration leads to non-anaphoricity, in spite of proximity.

Our strategy for creating the proximity sampled training data was as follows. For each pronoun, we determined the leftmost positive antecedent (within the extraction window), and included in the training data all morphologically compatible pairs (both positive and negative) in between. Thus, we get a configuration as follows (where P is a pronoun, N^+ is a correct antecedent of P (that is P is anaphoric or cataphoric to N^+), and N^- is a candidate that is not in a referential relation with P):

$$\ldots N^- \ldots N^- \ldots [N_1^+ \ldots N_2^- \ldots N_3^+ \ldots N_4^- \ldots P \ldots N_5^+] \ldots N^- \ldots$$

Left of the pronoun P, there is a number of NPs, some of which are coreferent with P, some other are not. N_1^+ is the leftmost NP that is an

	Baseline	Prox. sampled
# samples	146 153	84 731
# positive	27 649	27 649
# negative	118 504	57 082
ratio	1:4.287	1:2.065

Table 8.10: Comparison of baseline and proximity sampled training sets.

actual antecedent of P. No antecedents of P occur to the left of N_1^+. To the right of P, N_5^+ is the rightmost postcedent (a cataphoric relation).

Only the range of candidates between N_1^+ and N_5^+ is extracted as training samples, thus we get the combinations:

- $N_1^+ - P$
- $N_2^- - P$
- $N_3^+ - P$
- $N_4^- - P$
- $N_5^+ - P$

where N_2^- and N_4^- are included as negative training samples because they occur between N_1^+ and N_5^+.

Using this strategy, we create ten folds of training and set sets. In table 8.10, the unfiltered training sets and the proximity sampled training sets are compared.

As intended, the number of positive samples remains the same in both the baseline and the proximity sampled training set. The number of negative samples is nearly halved. This yields a positive-to-negative ratio of 1:2.065 as opposed to the baseline ratio of 1:4.287.

8.4.2 Vector-distance sampling

Vector-distance sampling is a variant of instance sampling that takes into account the distance between the feature vectors that represent positive examples and the feature vectors that represent negative examples in the

sample space of the memory-based learner. The usefulness of negative examples is determined by their proximity to the positive examples. The intuition here is that a negative example is especially useful if it is very close to positive examples. In terms of the search space, we are trying to concentrate on negative examples that are on the border to the positive class and remove the negative examples that are further apart.

For this sampling method, we trained the memory-based classifier on the positive examples only, using the optimal feature settings and the feature weights from the baseline experiment without sampling. Instead of the modified value difference metric, we however used the standard component-wise overlap metric.[9] Then we classified all the negative examples from the original training set against the positive examples and computed their distances to the closest positive example.

Based on this, we created two sets of sampled training data. For **close vector-distance sampling**, we selected only those negative examples that had a vector distance *smaller* than 0.002. The distance was empirically chosen so that a ratio of positive to negative examples would be reached of about 1:2, which was close to the ratios of the other sampling techniques. The actual ratio of the distance sampling method is 1:1.82. It turns out that there are no instances in the distance range between 0.002 and 0.004, but if we go beyond 0.004, we immediately almost double the number of negative instances.

In order to determine the discriminatory power of negative examples that are further away, we repeated the sampling and chose all the negative examples that had a distance greater than 0.002. We will refer to this method as **far vector-distance sampling**.

8.4.3 Incremental learning with the IB2 algorithm

This sampling variant is based on the incremental learning approach. It is based on the idea that it is not necessary to learn samples that can be classified correctly anyway, but only those that are problematic. One starts out

[9] MVDM depends on the class distribution but not on the individual components of the feature vector. Since there is only one class during training – the positive class – all samples will be assigned this class as well during classification. Their distance to the closest positive sample will always be 0 when using MVDM. The overlap metric computes vector distance based on the pairwise distance of the vector components and therefore does not exhibit MVDM's behavior.

with a limited set of samples to train on (in our case, the positive examples in the training data), and classifies new samples. Each sample that is classified incorrectly is then added to the training set. Thus, one incrementally updates the training set. This approach is called the IB2 algorithm, which is a modification of the standard IB1 memory-based learning algorithm suggested by Aha et al. (1991). Due to the fact that not all negative examples end up in the training set, this is approach is a form of sampling.

8.4.4 Random sampling

As mentioned before, the master set of training data contains 30 721 positive samples and 131 671 negative samples, which yields a ratio of 1:4.29. This is a ratio much lower than the one in Zhao and Ng's data, we therefore expect less strong effects for our data.

The general setup of the experiments is the same as for the baseline setup and the experiments with the other sampling methods, with some differences in detail to accommodate for possible artifacts of the randomization. As illustrated in figure 8.5, to perform ten-fold cross-validation, we split the master sample file in 10 parts, with $\frac{9}{10}$ of each part serving as training data, and the remaining $\frac{1}{10}$ serving as test data. But then, we additionally performed instance sampling on the training sets of each fold. We prepared eight different sets of training data with ratios of 1:1, 1:1.5, 1:1.75, 1:2, 1:2.25, 1:2.5, 1:3 and 1:4. For each set, we randomly removed negative pairs until the desired ratio was reached. In order to eliminate randomization artifacts, we created 10 different training sets for each target ratio.

We extracted the test sets from the original master instance file, without any sampling. This ensures that the performance results remain comparable over all folds and experiments.

For each fold, we trained TiMBL on one of the randomized sampled training sets. We then averaged the results of the ten randomized sets for each ratio to exclude artifacts of the randomization from the evaluation process, thus ending up with one performance value of the resolver for each of the ratios.

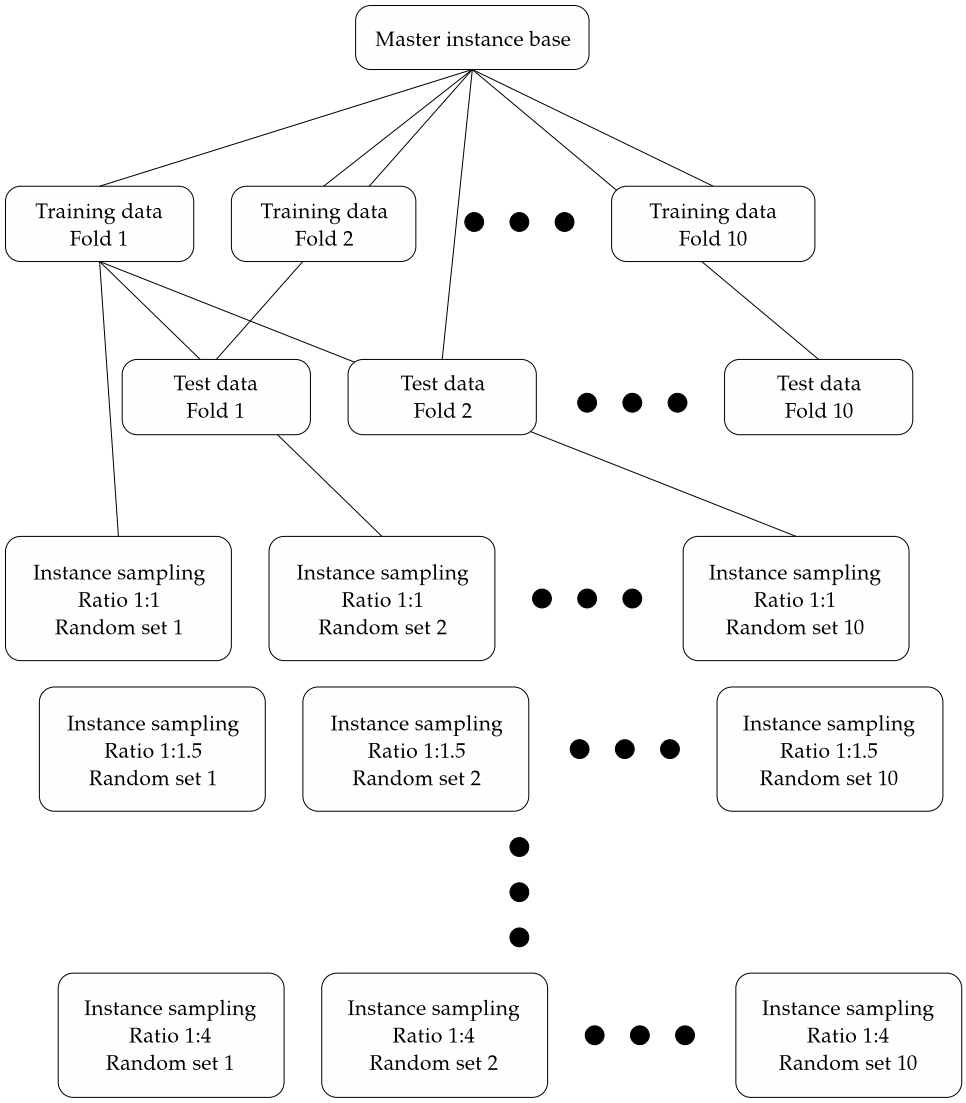


Figure 8.5: Training data sets for experiments with random sampling

	Ratio	Prec.	Rec.	F
Baseline (no sampling)	1:4.29	**0.664**	0.457	0.541
Proximity sampling	1:2.1	0.511	0.707	0.593
Close vector-distance sampling	1:1.82	0.504	0.547	0.525
Far distance sampling	1: 2.47	0.458	**0.801**	0.583
Incremental Learning	1:0.96	0.592	0.511	0.547
Random sampling	1:1	0.479	0.783	0.593
	1:1.5	0.502	0.751	0.602
	1:1.75	0.521	0.720	**0.604**
	1:2	0.542	0.683	**0.604**
	1:2.25	0.552	0.662	0.602
	1:2.5	0.567	0.632	0.598
	1:3	0.598	0.570	0.584
	1:4	0.653	0.477	0.552

Table 8.11: Results of the instance sampling experiments.

8.4.5 Experiments and results

The technical setup of the experiments for all methods was essentially the same as for random sampling. We extracted ten folds from the master instance base, with $\frac{9}{10}$ of each fold reserved for the training data, and the remaining $\frac{1}{10}$ for the test data. Then, we sampled the training sets with each of the sampling methods. We did not sample the test sets, thus they are equal for each of the four experiments and the baseline experiment. This ensures the comparability of the results across all experiments. An overview of the results is shown in table 8.11.

The results show that instance sampling strongly affects the performance of the memory-based resolver. The baseline achieves the highest precision (0.664), but at the same the recall is the lowest of all experiments (0.457). This corroborates the hypothesis that instance sampling affects the classifier bias towards a more lenient classification strategy.

Although the effects of random sampling are not as strong as in Zhao and Ng's experiments, they are still clearly visible, as illustrated in table 8.11 and in figure 8.6. The baseline experiments correspond to a ratio of 1:4.29, which is the rightmost data point in the chart. The f-measure of

the performance of the baseline experiments is the lowest in the complete series, which is to be attributed to the low recall of 0.457. Starting with the ratio 1:1 and ending with 1:4.29, the values of precision rise monotonically, while recall drops. Thus the experiments show the behavior of a classifier with increasingly conservative bias. With the training set of ratio 1:1, i.e. a balanced training set, the classifier is most likely to classify pairs as anaphoric, which is reflected by the high recall of 78.3%. However, many of these pairs are not classified correctly, which the low precision of 47.9% is an indication for. As the curve for f-measure shows, the performance optimum (with respect to f-measure) is reached at a ratio between 1:1.75 and 1:2. This version of the training set is less balanced and leads the classifier to a slightly more conservative classification strategy, resulting in an f-measure of 0.605, and an increased precision of 0.538. Recall decreases to 0.690. With increasing number of negative samples, recall starts to drop steeply, resulting in a substantial loss of overall performance. This loss cannot be compensated by precision, which does rise, but at a much slower rate than the decrease of recall.

Proximity sampling, i.e. reducing the negative examples to the ones found between the pronoun and its correct antecedent increases recall by 25 percent points, but it also decreases precision by approximately 15 percent points, resulting in an increase of the F-score of 5 percent points.

Close vector-distance sampling is less successful: There is a sightly higher decrease in precision, but recall much lower at 0.547 than for proximity sampling.

The results for far vector-distance sample are surprising. Recall reaches the highest value of all experiments (0.801), however precision is very low (0.458), resulting in an average F-score of 0.583. This shows that clearly separated training samples help to significantly increase recall.

The incremental learning approach IB2 presents the next surprise: The sampling ratio is the lowest of all experiments (1:0.96), which should result in high recall and low precision, but the opposite is the case: With 0.592, precision is higher than for all other sampling approaches except for random sampling with almost the complete set of negative instances (1:4). Correspondingly, recall is lower (0.511) than for most other sampling approaches, with the same exception. And while the F-score is fairly stable across the 10 folds, precision and recall vary considerably more than in all

Pronoun type	Prec	Rec	F	# positive	# negative	Ratio
reflexive	0.895	0.872	0.883	2282	7702	1:3.38
personal	0.621	0.496	0.552	18393	60162	1:3.27
possessive	0.691	0.291	0.409	10046	63807	1:6.35

Table 8.12: Performance of the hybrid resolver by pronoun type on the baseline training data.

experiments: The highest precision was reached for fold 7 with 0.642, the lowest precision was 0.544 for fold 4. Recall varied between 0.549 for fold 2 and 0.498 for fold 4.

To summarize, the results show that instance sampling methods do affect the performance of the classifier. All outcomes of all experiments show the effect of the sampling ratio on the classifier: The more the negative samples outweigh the positive samples, the stronger the conservative bias of the classifier. However, the results cannot only attributed to a reduced bias. The non-random sampling methods far vector-distance sampling and incremental learning clearly show that additional information about the nature of the samples has a strong influence on the classifier results. The sampling ratio of training sets generated by incremental learning is 1:0.96, and, compared to the scale in random sampling, should therefore lead to very low precision and very high recall. However as described above, this is not the case. Thus if efficiency is a concern, a sampling method such as incremental learning can be beneficial as it is capable of keeping the instance base small while preserving satisfactory performance. In memory-based approaches, the size of the instance base is directly correlated to the processing time needed to classify new samples. The results show that for a machine-learning-based approach, the best strategy is to pick samples from all areas of the search space, as done in random sampling, instead of restricting the areas based on linguistic considerations.

8.4.6 Evaluation of random sampling by pronoun type

In addition to examining the influence of random sampling on the overall performance of our resolution system, it is also interesting to assess the strategy's influence on the individual resolution quality of the three dif-

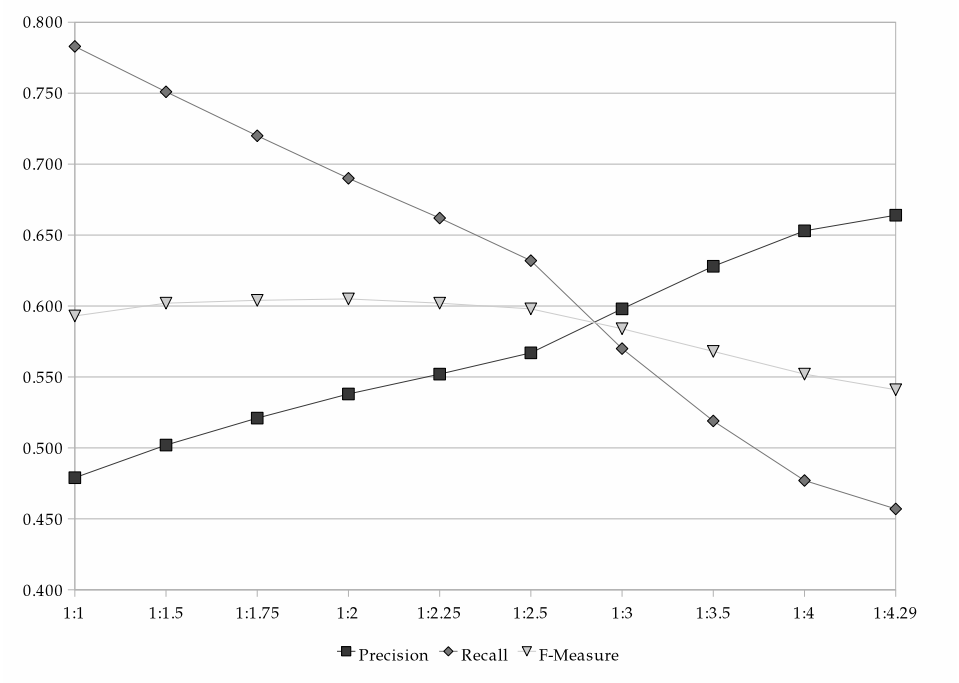


Figure 8.6: Results of instance sampling with different ratios along the X-axis.

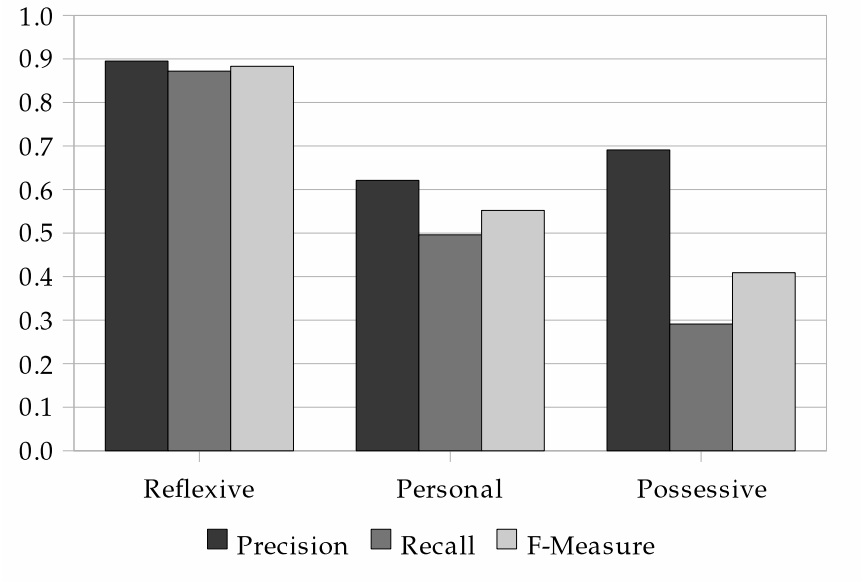


Figure 8.7: Performance of the hybrid resolver by pronoun type on the baseline training data.

ferent types of pronouns. To this end, we computed an average per-ratio performance profile broken down into the three types of pronouns considered by the resolver: personal pronouns, reflexive pronouns, and possessive pronouns. For each of the ten folds, we extracted from the TiMBL-classified test data those pairs that matched the respective pronoun type and then computed precision, recall, and f-measure in the usual fashion on the sets such produced.

Figure 8.7 and table 8.12 show the performance of the resolver by pronoun type on the baseline data (with the default ratio of 1:4.29 of positive versus negative samples in the training data). It is obvious that there are great differences with the individual pronoun types. On reflexive pronouns, the system performs very well at an f-measure of 0.883, with values of precision (0.895) and recall (0.872) that are quite balanced.

Personal pronouns are next, with an f-measure of 0.552 which is substantially lower. The performance is worst on possessive pronouns. Here, the system only achieves an f-measure of 0.409. The difference of precision and recall is much stronger than with the other two types of pronouns: recall is lower than precision by 40 points of percentage. Table 8.12 lists the number of positive and negative pairs of the individual pronoun types, and the ratio of positive and negative instances.

For possessive pronouns, the ratio is extremely skewed towards negative instances (1:6.35). This corresponds directly with the behavior of the classifier: it has a strong bias towards negative classification – leading to the very low number of pairs actually resolved. The order of magnitude of the ratio for personal pronouns (1:3.27) is similar to that of the overall cross-pronoun ratio (see table 8.11) – and so is the performance of the classifier on this pronoun type.

Reflexive pronouns however do not fit into this pattern at all. The ratio of positive and negative instances is only a little bit higher than that of personal pronouns (1:3.38), while there are much less pairs involving reflexive pronouns in the training set in total. Given this, we would expect a performance similar to that of personal pronouns. However, the resolution quality is far superior. The reason for this result can be explained when looking at the syntactic domains the three types of pronouns can occur in. Reflexive pronouns are strongly restricted. Their antecedent may only occur in the same local syntactic domain, and it must be a coargument of the

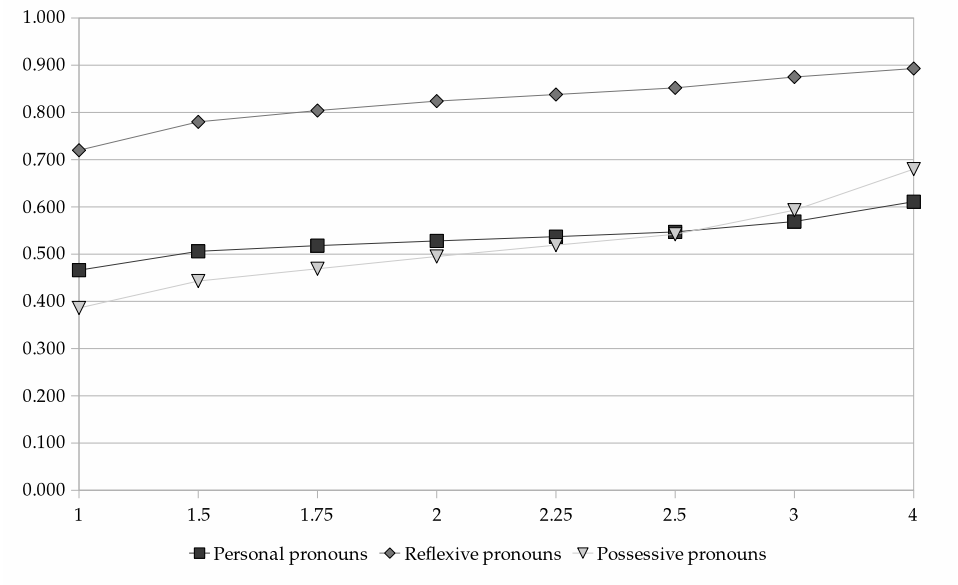


Figure 8.8: Precision by pronoun type and ratio

pronoun. Even with a knowledge-poor approach, this configuration can be enforced already with a very simple rule-based syntactic filter, as the one we use in our approach. The syntactic domain of personal pronouns is less restricted, even more so since they may enter inter-sentence referential relations for which sentence-based syntactic filters cannot be applied. Nevertheless, they are still likely to occur in argument positions with grammatical functions parallel to their antecedents. Thus, while the syntactic domain of personal pronouns is much less restricted than the domain of reflexive pronouns, their distribution is still defined such that sufficiently informative features can be derived that the classifier will benefit from. Compared to the other two types of pronouns, the syntactic domain of possessive pronouns is most unrestricted. The syntactic constraints that apply to pronouns that occur in argument positions cannot be applied to possessive pronouns as they usually do not occur in an argument position, but rather as attributive modifiers.

In order to find out how the performance on individual pronoun types is sensitive to sampling, we evaluated the output of the resolver separately for each pronoun type and examined the development of precision, recall, and f-measure. It shows that the differences in performance seen in the baseline experiments continue to be evident in the sampling experiments.

Ratio	Reflexive	Personal	Possessive
1:1	0.720	0.466	0.386
1:1.5	0.780	0.506	0.433
1:1.75	0.804	0.518	0.469
1:2	0.824	0.528	0.495
1:2.25	0.838	0.537	0.519
1:2.5	0.852	0.547	0.542
1:3	0.875	0.569	0.593
1:4	0.893	0.611	0.680

Table 8.13: Precision of pronouns by type

Figure 8.8 illustrates the development of precision for the three pronoun types (table 8.13 lists the accurate numbers). For all sampling ratios, the precision on reflexive pronouns is far better than for the other two types of pronouns. It starts at 72 % for a ratio of 1:1 and rises to 87.2 % for the ratio 1:4 with the increasing conservative bias of the classifier. At the same time, recall, shown in figure 8.9 and table 8.14, remains fairly constant, dropping from 94.1 % for the ratio 1:1 to 87.2 % for the ratio 1:4. This indicates that for reflexive pronouns, a conservative classification strategy is beneficial: The moderate loss in recall means that in spite of the increasing negative evidence, the number of pairs that the classifier looses is fairly small, while the increasing precision reflects the improving resolution quality. This improvement towards more conservative classification is reflected by the f-measure, which increases by 6.6 points of percentage between the 1:1 and 1:4 ratios, as shown in figure 8.10 and table 8.15.

The situation is very different for personal and possessive pronouns. Most striking is the substantial loss of recall for both pronoun types with the increasing frequency of negative training samples (see figure 8.9 and table 8.15). Recall for personal pronouns goes down by 32.7 points of percentage. The drop is even higher for possessives - here it loses 43.5 points of percentage between the ratios 1:1 and 1:4. This cannot be made up for by the rising precision which behaves similarly for both personal and possessive pronouns. Precision increases by 14.5 points of percentage for personal pronouns, which is roughly in the same order of magnitude as the increase in precision for reflexive pronouns (albeit at a lower level). The increase

Ratio	Reflexive	Personal	Possessive
1:1	0.941	0.850	0.739
1:1.5	0.924	0.791	0.673
1:1.75	0.913	0.767	0.589
1:2	0.903	0.745	0.542
1:2.25	0.896	0.721	0.501
1:2.5	0.888	0.694	0.461
1:3	0.881	0.635	0.380
1:4	0.872	0.523	0.304

Table 8.14: Recall of pronouns by type

Ratio	Reflexive	Personal	Possessive
1:1	0.816	0.602	0.507
1:1.5	0.846	0.617	0.522
1:1.75	0.855	0.618	0.522
1:2	0.862	0.618	0.518
1:2.25	0.866	0.615	0.510
1:2.5	0.870	0.612	0.498
1:3	0.878	0.600	0.463
1:4	0.882	0.564	0.421

Table 8.15: F-measure of pronouns by type

in quality of possessive pronouns is larger, precision rises by 29.4 points of percentage. For ratios greater than 1:2.5, the precision of possessives even exceeds that of personal pronouns.

The total situation is reflected by the f-measure, as shown in figure 8.10 and in table 8.15. For reflexive pronouns, f-measure rises with more conservative classification, while for the other two types of pronouns, the f-measure takes a maximum between 1:1.75 and 1:2, and then drops again. The separate evaluation of the pronoun types shows that the ranking of performance remains the same for all ratios, with possessive pronouns most sensitive to sampling.

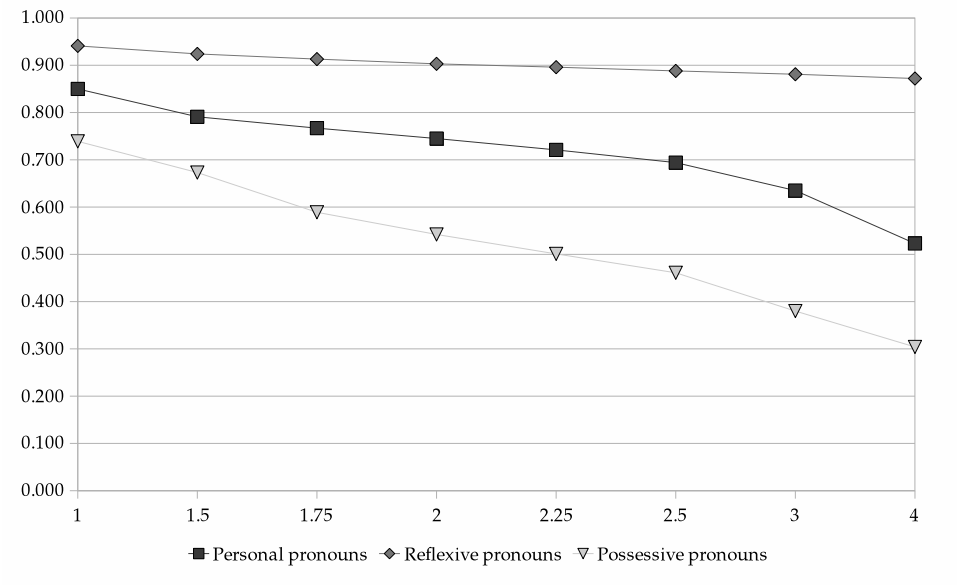


Figure 8.9: Recall by pronoun type and ratio

8.5 Summary

In this chapter, we presented a hybrid system for pronoun resolution, which combines rule-based and machine-learning-based methods in three modules:

- a rule-based morphological prefilter which removes pairs of pronouns and potential antecedents that do not agree in number and gender. It is effective – 54% of the pairs that are in fact non-anaphoric are removed.
- a machine-learning-based core resolution module using the TiMBL memory-based classifier.
- a rule-based heuristic postfilter that (i) selects the leftmost antecedent when multiple antecedents were selected for a pronoun and (ii) picks the closest morphologically compatible subject when no antecedent could be found by the resolution module.

The results of this system show that a combination of components based on simple linguistic rules for filtering steps with a machine-learning-based core anaphora resolver is well suited to construct a system which reaches

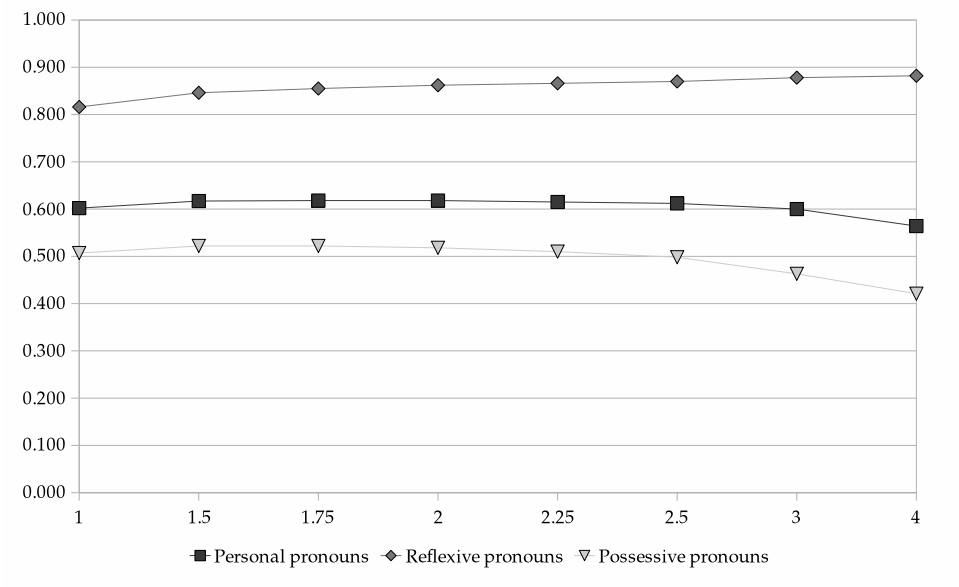


Figure 8.10: F-measure by pronoun type and ratio

performance levels of that of a manually tailored, completely rule-based system (Hinrichs et al., 2005b).

In a further research topic, we examined the influence of the structure of the training data, more specifically, the distribution of positive and negative samples on the behavior of the classifier. The first result is that the more skewed the data toward negative samples, the more conservative the classifier behaves, i.e. it is biased towards negative classification. This results in high precision, but low recall. *Instance sampling* is a method of altering the skewedness of the training data in a controlled fashion towards an increased weight of positive samples. We experimented with four different sampling methods and found noticeable effects of instance sampling. While for random sampling, the correlation of sample ratio and classifier bias is obvious, this is not the only effect of instance sampling on the data, as the results with the incremental learning algorithm show. Although the training set created using incremental learning contains only half of the negative samples as the optimal setting for random sampling, precision for incremental learning reaches values that are equivalent to conservative training sets with roughly 3 times the number of negative samples in random sampling.

Two conclusions are to be drawn:

- Comparing the results of all sampling methods, random sampling reaches the best tradeoff between precision and recall when compared to the other sampling methods. Thus, a sample set with samples *from all areas of the sample space* is the most informative and to be preferred over sample sets constructed from restricted search spaces.
- Specific kinds of instance sampling, particularly incremental learning, are useful to optimize a training set for size without losing too much performance. With memory-based approaches this can be especially important, as the size of the training set directly affects the classifier's memory usage and processing time.

Chapter 9

Semantics for Pronoun Resolution

In the introduction to this thesis, we described the concept of cohesion, and we called cohesion the "semantic glue" that renders a sequence of sentences into a meaningful whole – a text. A relation of cohesion holds between two elements of a text, and it can be expressed with very different linguistic means, for example lexically, phonologically, or in syntax. But regardless of the stratum on which the relation is linguistically realized, it finally ends up as a purely semantic relation.

Since anaphora is a special form of cohesion, it is of course also a semantic relation. Anaphora may be subject to certain syntactic restrictions when it occurs between a pronoun and a noun phrase within the same sentence, as we saw in chapter 2 in the section on binding theory. But its core properties are on the level of discourse.

Jerry Hobbs closes his seminal paper on resolving pronominal anaphora (Hobbs, 1978) with an outline of a complete system for resolving pronouns, which operates *exclusively* on semantic grounds. Hobbs himself considers his syntactic algorithm for pronoun resolution, which we introduced in chapter 3, only a *fallback step*, should the semantic part of the system fail to find an antecedent.

Hobbs' work marks the beginning of longstanding research on algorithms to solving the problem of anaphora resolution in computational linguistics. At first sight, it may therefore be surprising that until very recently most systems that were developed to run on computers only imple-

mented weak notions of semantics. Aone and Bennett (1995) use a semantic class feature for definite NPs which they do not elaborate on. Soon et al. (2001) use concept lookup in WordNet, and a surface string match feature. Strube et al. (2002) experiment with three features: string identity, substring match, and the minimum edit distance between definite NPs. They explicitly state that their features do not apply to pronouns. Ng and Cardie (2001) use a word net class feature. As a matter of fact, none of them consider semantic information in the pronoun resolution process.

The reasons for the virtual absence of semantic processing from approaches to pronoun resolution in computational linguistics are easily identified, and similar thoughts have been brought up in several places in this thesis: In order to be expressed as an algorithm that can be implemented on a computer, any approach must have access to sources of data of sufficient size and reliability. For syntax as well as morphology, parsers and taggers are available today that provide analyses of good accuracy, which are used to annotate data on a scale large enough to be applicable for computational approaches. Furthermore, over the last years corpora have been made available for numerous languages that contain manual annotations for syntax and morphology. They provide data of the necessary quality to train and verify the computational approaches on the levels of syntax and morphology. To this date however, neither manually created resources nor computerized analysis tools are available that would provide *semantic* analyses of both the quantity and quality necessary, especially for pronoun resolution.

The situation is slightly different for the task of resolving definite NPs. In recent years, a number of studies were introduced that aim to incorporate more extensive semantic knowledge sources. One strategy is to classify definite NPs and their potential antecedents into broader semantic classes such as PERSON, TIME, LOCATION, OBJECT, and so on.

Soon et al. (2001) experiment with the hypernymy hierarchy of the Princeton WordNet (Miller et al., 1988) for finding such semantic classes. It is also possible to obtain semantic class information from large text corpora such as Wikipedia or the BBN Entity Type Corpus (Weischedel and Brunstein, 2005). In Wikipedia, articles are arranged in an elaborate, extensible taxonomy that can be taken advantage of to arrive at the semantic class information in question. The BBN Entity Type Corpus

contains inline semantic class annotation of markables.

Ponzetto and Strube (2006) combine a semantic similarity measure of two NPs based on WordNet with a measure of semantic relatedness derived from the taxonomy that is contained in Wikipedia. Furthermore, they check semantic compatibility of two NPs by testing whether two *interlinked* articles exist in Wikipedia whose headlines contain a head word of the markable. Two articles are interlinked either if there exists a direct link between the two, or by means of a redirection or disambiguation page.

Ng (2007) finally reports on a statistical semantic tagger that is trained on the BBN Entity Type Corpus which discriminates among six broadly defined semantic class PERSON, ORGANIZATION, FACILITY, GEO-POLITICAL ENTITY, and LOCATION, which are taken from the ACE Phase 2 coreference corpus.

For pronouns however, the situation is more complex. Pronouns by themselves are semantically empty. Specifically, unlike a definite NP, whose surface form may hint at its semantics, no such hint can be found in the surface form of a pronoun.

Our main concern in this chapter is the question whether information about the semantics of pronouns and candidate antecedents can help improve the performance of the resolver. To this end, we will first address the issue of how to make lexical semantics applicable to pronouns. We will then develop a notion of semantic compatibility between a pronoun and a candidate antecedent that can be integrated in our hybrid pronoun resolution system. We will close this chapter with the evaluation of the results and their discussion.

9.1 Shortcomings of syntactic features

Prior to moving on to the discussion of our data-driven approach of gathering semantic information, we will first address the question what the shortcomings of the usage of traditional syntactic features are.

An alternative to the traditional view of the relevant information as features that guide the resolution process is to conceive them as a set of weak constraints that restrict the search space of potential antecedents, in a sense similar to the status of constraints in Optimality Theory (such as the set of constraints that Beaver (2004) suggests for expressing Centering Theory,

Factor type	Initial weight
Sentence recency	100
Subject emphasis	80
Existential emphasis	70
Object emphasis	50
Indirect object and oblique complement analysis	40
Head noun emphasis	80
Non-adverbial emphasis	50

Table 9.1: Ranked salience hierarchy by Lappin and Leass (1994), used in their Resolution of Anaphora Procedure (see chapter 6)

see also section 2.6 in chapter 2). Unlike a hard constraint which would immediately rule out a violating instance (such as a free reflexive pronoun violating binding principle A), weak constraints may be violated. A *ranking* of these constraints states the relative importance of the constraints and how severe it is for a constraint to be violated.

Provided a proper ranking, these constraints should make consistent predictions about what candidate is to be selected as the antecedent of a pronoun, i. e. their predictions should be *stable*. However, as Hinrichs and Wunsch (2009) extensively discuss for English, morphosyntactic features are *not stable:* Even with a carefully crafted ranking of features such as the one worked out in the salience hierarchy by Lappin and Leass (1994) shown in table 9.1, it is fairly easy to construct minimal pairs that trigger false predictions:

(1) Vincent removed **the diskette** from **the computer** and then switched **it** off.

(2) Vincent removed **the diskette** from **the computer** and then copied **it**.

The two constraints that compete in example (1) and (2) are *recency* and *object emphasis*. In both sentences, the set of candidate antecedents for the pronoun *it* contains the NPs *the diskette* and *the computer*. As depicted in table 9.1, *recency* is ranked higher than *object emphasis*. In example (1), the pronoun *it* refers to *the computer*, which is the closer NP, as correctly pre-

dicted by the *recency* feature. However, in example (2), which is syntactically completely equal to example (1), the correct antecedent would be *the diskette*. But again, due to the ranking, *recency* prevails over *object emphasis*, therefore *the computer* again is selected as the antecedent. Moreover, if the ranking of the two features was flipped, the solution would *still not be stable*. In this case, object emphasis would prevail in both cases, now correctly resolving (2), but not (1). Obviously, the relevant syntactic features cannot fully capture the phenomenon that is in question in these examples.

9.2 Semantic features

We saw in the previous section that features that are solely based on syntax may fail to select the correct antecedent even when ranked by a carefully crafted salience hierarchy. Inspecting examples (1) and (2) in the previous section more closely, the reason for the inability of the syntactic features to make correct predictions is that the determining factor in these sentences is actually the *semantics of the verb*: In the first example, the verb is *switch off*, which is an action that can typically be carried out on appliances like computers. In the second example, the verb is *copy*. Actions of copying can be applied to *diskettes*, but not to *computers*.

The property of a verb to restrict the kinds of entities that it accepts as one of its argument is known in the literature by the term *selectional preference*. Verbs impose strong restrictions on the kinds of entities that may occur as their complements: Nouns that denote types of *food* typically occur as objects of verbs like *cook* or *eat*. Verbs that belong to the semantic field of *hear* may select objects like *music*, *opinion*, or *word*. It is extremely unlikely that *words* are *cooked*, and *cucumbers* are *heard*.

In the examples discussed in the previous section (repeated here as (3) and (4)), the two relevant verbs are *switch off* and *copy*. The set of candidate antecedents of the pronoun contains two elements: *computer* and *diskette*.

(3) Vincent removed the diskette from **the computer** and then *switched* **it** *off*.

(4) Vincent removed **the diskette** from the computer and then *copied* **it**.

The selectional preferences of *switch off* in (3) consist of the set of NPs that denote entities in the real world that are electrically powered and that pro-

vide a switch to turn them on and off. A *computer* is indeed an object that can be turned on or off. This means that it is selectionally preferred by *switch off*, while *diskette* is not. With the additional semantic constraint that the antecedent of the pronoun must be a member of the selectional preference set of the verb, the pronoun *it* can be correctly resolved to the antecedent *computer*. Admittedly, this is not overly exciting, as the recency positional feature would have arrived at the same result.

However in example (4), matters are different: The selectional preferences of the verb *copy* include objects that can be reproduced using some physical action. A *computer* is certainly not an object that can straightforwardly be reproduced, while a *diskette* is. Thus, *diskette* is in the set of selectionally preferred entities, while *computer* is not. Therefore, if the selectional preference constraint is applied here, *computer* is ruled out, while *diskette* is correctly selected as the antecedent, thus overruling the syntactic constraints.

9.3 Data-driven extraction of selectional preferences

Selectional preferences are lexical properties of a word. In order to make them accessible to a computer program such as the hybrid pronoun resolver, they must be represented as explicit information, so that the program can query the necessary information for each verb it encounters. In principle, the easiest way to achieve this would be a lexicon. This lexicon should satisfy two requirements. Firstly, it should contain for each verb the information what semantic classes are acceptable in each of the verb's argument positions. Secondly, it should be large-coverage, and provide the necessary information for most of the verbs to be encountered in the target domain. Creating a lexicon of this size by hand would be an an enormous, if not impossible, task. Instead of manually creating a lexicon, data-driven approaches are employed in computational linguistics that extract databases of selectional preferences from very large corpora.

The selectional preferences of the verbs are not visible in a text. Still, they do apply when the text is created: A noun phrase that occurs as the direct object of the verb must obey its selectional preferences for the whole construction to be sensible. For example the sentence *"the cook heard the cucumbers"*, which is rather obscure, will probably hardly occur anywhere

except right here in this sentence, since the direct object *cucumber* does not belong to the selectional preferences of the verb *hear*. The sentence *"the cook tasted the cucumbers"* is much more likely, since the semantic fields that are admitted by the selectional preferences of *taste* include that of food.

Several approaches to extracting selectional preferences have been proposed in the literature, requiring different degrees of prior knowledge. A measure that is well suited for determining the collocational strength of verb-object pairs is the log-likelihood ratio (Dunning, 1993). Rooth (1998) introduces *Latent Semantic Clustering*, an unsupervised statistical approach which computes smoothed clusters of verbs and nouns that represent the selectional preferences. Resnik (1993) uses a mapping of nouns to WordNet concepts to compute the *selectional preference strength* of a verb. Abe and Li (1996) use a tree cut model computed on the basis of the information-theoretic minimum description length principle to determine selectional preferences of adequate degree of abstraction. Wagner (2005) combines information theoretic approaches of concept mapping with clustering approaches to learn thematic role relations for the extension of lexical-semantic networks.

The first step of our data-driven approach to extract selectional preferences is therefore to collect all verbs and their arguments from the text base that serves as the data source. The result of this will be a huge list of pairs of verbs and nouns that occur together. Next, it must be counted how often a specific verb-noun pair occurs in the list, which leaves us with a frequency profile of the co-occurrence of every verb in the corpus and its arguments. We used the TüPP-D/Z treebank (see section 5.4) as the data basis for extracting pairs. TüPP-D/Z does not contain grammatical functions, we therefore set up a distributed computer system for annotating the whole corpus with grammatical functions. On the individual compute nodes, we deployed the KaRoPars system (Müller, 2004a) with a rule-based grammar for grammatical function annotation.

TüPP-D/Z is suited quite well for the task of extracting selectional preferences. Its automatic syntactic annotation is most reliable on the level of noun chunks and on the layer of topological fields. The structure of the topological fields delimits the left and right sentence brackets, which contain the verbal material. Thus, the annotation of topological fields with good quality is a prerequisite for reliably identifying the verbs in a sentence

	Prec.	**Rec.**	$\mathbf{F}_{\beta=1}$
LK	97.39%	96.08%	96.73
VC	90.95%	94.49%	92.68
NC	87.42%	88.47%	87.94

Table 9.2: Performance of the TüPP-D/Z automatic parser

and extracting them correctly. In the same way, the quality of the annotation of noun chunks influences the extraction of the nominal arguments. Müller (2007) reports for the annotation of the left and right sentence brackets and for the noun chunks in TüPP-D/Z the results summarized in table 9.2. They are determined by comparing the automatic annotation in the TüPP-D/Z treebank with the manual gold annotations in the TüBa-D/Z treebank (using only those sections that are present in both treebanks). The performance of the end-to-end automatic system (i.e. automatic segmentation, POS-tagging and parsing) is satisfactory. The parser performs best on the left sentence bracket (LK), reaching an f-measure of 96.73. It is slightly worse on right sentence brackets (VC, verb complex – 92.68). Noun chunks are less reliable, here the parser only achieves an f-measure of 87.94.[1]

For the present experiments, we extracted the selectional preferences of the verbs on their subjects and accusative objects. We did not include selectional preferences for dative or genitive objects, neither did we consider prepositional objects. There are two reasons for this.

Firstly, as illustrated in table 9.3, arguments are distributed very unequally with respect to their grammatical function. Subjects (which are assigned the GF label `ON`) occur most often by far, followed by accusative objects (grammatical function `OA`). Dative objects (`OD`) occur much more seldom, and genitive objects (`OG`) are virtually not present, compared to the frequency of the other grammatical functions. As stated before, the validity of the results of clustering vitally depends on sufficient amounts of raw data. For subjects and accusative objects, enough data is available. For dative and genitive objects, this is not the case.

[1]Due to the differing design decisions in the grammar that drives the parser for annotating the TüPP-D/Z corpus and the TüBa-D/Z annotation guidelines (with respect to postmodification of NPs for example), it should be noted that this evaluation strategy yields partly unfair results.

GF	Frequency
ON	13 110 987
OA	8 742 410
OD	908 717
OG	4 320

Table 9.3: Distribution of grammatical functions in TüPP-D/Z

	Prec.	Rec.	$F_{\beta=1}$
ON	90.93%	91.27%	91.10
OA	82.94%	82.77%	82.86
PRED	78.53%	72.87%	75.59
OD	83.66%	59.40%	69.47
OS	76.92%	60.67%	67.83
OPP	72.06%	48.50%	57.98
OG	100%	7.69%	14.29
overall	85.52%	80.02%	82.68

Table 9.4: Evaluation of the KaRoPars GF annotation component (Müller, 2007)

The extraction module furthermore depends on the quality of the annotation of grammatical functions proper. Müller (2007)[2] reports the figures given in table 9.4 for the GF annotation module of the KaRoPars system. The annotation quality mirrors the ranking in frequency. The grammatical function `ON` is annotated with the highest f-measure, followed by accusative objects. Dative objects are significantly worse. The values for genitive objects are nearly non-interpretable, as argued by Müller, since in the test set he uses, only one genitive object occurs at all.

Viewed in combination, subjects and accusative objects meet the requirements of (a) occurrence with sufficient frequency and (b) sufficient quality in the TüPP-D/Z corpus, while other complements or adjuncts occur with too low frequencies or inadequate annotation quality. We therefore restrict the extraction of selectional preferences to subjects and accusative objects.

[2] p. 299 ff

9.3.1 Extraction of verb-subject and verb-object pairs

This section will outline the algorithm used to extract verb-subject and verb-object pairs from the TüPP-D/Z corpus. The algorithm locates the main verb and looks for the subject and its direct object in a clause. The algorithm can detect constructions in eventive passive voice, in which case the subject of the sentence is treated as if it was the direct object of an active sentence, i.e. extracted in the object position of a verb-object pair. We will describe the passive detection algorithm in the next section.

In the following, we will present the extraction algorithm in detail.

1. *Process the whole corpus sentence by sentence.*

2. *Within a sentence, process all clauses in a left to right, top to bottom order.*
   The algorithm treats embedded clauses independently of a matrix clause, i.e. verbs, subjects and direct objects are extracted from embedded clauses just the same as from a main clause.

3. *From a clause, extract the noun chunks with grammatical function* `ON` *and* `OA`.
   These noun chunks are the arguments of the verb.

4. *Determine whether the clause is in passive voice*.
   If so, the subject of the sentence is extracted as direct object as if the sentence was in active voice. No verb-subject pairs are extracted from a passive sentence even if the agent is specified in a prepositional phrase. See section 9.3.2 below for a detailed of the passive detection algorithm.

5. *Extract embedded noun chunks only if they have the same grammatical function as the embedding NP.*
   This is a filter that ensures that only heads of NPs are extracted. Note that this step does not apply if no embedded chunk is annotated with a grammatical function – but see step 6.

6. *Remove word tokens from the yield of the extracted NP with parts of speech other than the permitted parts of speech.*
   The permitted parts of speech are: NN (common noun), NE (proper noun), PDS (substitutive demonstrative pronoun), PIS (substitutive indefinite pronoun), PPER (personal pronoun), PPOSS (substitutive

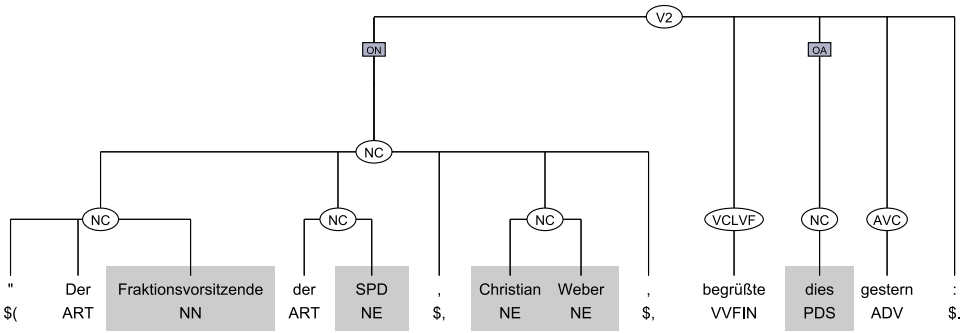


Figure 9.1: Only the highlighted NPs pass the POS filter.

possessive pronoun), PRELS (substitutive relative pronoun), PRF (reflexive pronoun), or PWS (substitutive interrogative pronoun).

The purpose of this filter is to remove function words and modifiers from noun chunks that are not explicitly marked otherwise. Figure 9.1 shows a relevant clause from TüPP-D/Z. The main verb is *begrüßte*, the NC *"Der Fraktionsvorsitzende der SPD, Christian Weber"* is marked ON, and the demonstrative pronoun *"dies"* is the accusative object. In the subject NC, the embedded noun chunks are not annotated any further, especially, there is no information about what is the head. Only the nouns *"Fraktionsvorsitzende/NN"*, *"SPD/NE"*, [3] *"Christian/NE"*, *"Weber/NE"* pass the filter. The pronoun *"dies/PDS"* passes the filter as well.

7. *Locate the main verb of the clause.*
   The main verb occurs either in the left sentence bracket or in the right sentence bracket, depending on whether the clause is a verb-first, verb-second, or verb-last clause.

   Auxiliary verbs are dropped. No pairs are extracted that contain an auxiliary verb.[4]

[3] The filter could be extended to remove adjuncts in genitive as well. However, the case annotation in TüPP-D/Z is not fully disambiguated. Relying on this information would harm the precision of the filter.

[4] In sentences such as *"Er ist krank"* the verb *ist* is annotated as auxiliary (due to the automatic tagging) even though it is a full verb. However, verbs like *sein, haben* do impose very weak selectional preferences on their arguments which is why we chose *not* to extract pairs involving these verbs.

8. *Add the new pair to the list.*
   If multiple tokens are extracted as in 6. above, one pair for each token is added to the list.

The result of this procedure is a list that contains all verbs in the corpus that have at least one argument, and the arguments themselves.

9.3.2 Passive detection

The purpose of passivization is to shift focus from the agent to the patient. In a sentence in active voice, the agent is usually realized in the position of the subject, and the patient in the position of the direct object:

(5) Die AWO$_{subj/agent}$ bezahlt das Altenheim$_{acc\text{-}obj/patient}$ in Danzig.
The AWO pays the home for the elderly in Gdansk.

'The AWO pays for the home for the elderly in Gdansk.'

In passive voice, the patient is moved to the subject position, while the agent is now realized as a prepositional adjunct:

(6) Das Altenheim$_{subj/patient}$ in Danzig wird von der AWO$_{prep/agent}$
The home for the elderly in Gdansk is by the AWO
bezahlt.
paid.

'The home for the elderly in Gdansk is paid for by the AWO.'

Generally speaking, we can establish that the selectional preferences that are imposed on the direct object in active sentences are imposed on the subject in passive sentences.[5] Therefore, the pair that is extracted as a verb-object pair from *both* examples above is the pair *bezahlen – Altenheim*.

The quality of data-driven approaches crucially depends on the amount of available data. With the passive detection routine we extend our verb-noun extraction system to sentences in eventive passive voice. Since our goal is just to harvest additional reliable verb-noun pairs, we do not implement a full treatment of German passive. We specifically refrain from

[5] Actually, selectional preferences of verbs are restrictions on the semantic classes of their agents and patients, and not of their syntactic arguments. If seen from this perspective, the active/passive-distinction becomes irrelevant.

VF	LK	MF	RK
Der Vize-	wird/VAFIN	in seinem	bestätigt/VVPP
präsident	wurde/VAFIN	Amt	bestätigt/VVPP
	wird/VAFIN		bestätigt/VVPP werden/VAINF
	wird/VAFIN		bestätigt/VVPP worden/VVPP sein/VAINF
	ist/VAFIN		bestätigt/VVPP worden/VVPP
	kann/VMFIN		bestätigt/VVPP werden/VAINF

Table 9.5: Patterns of passive that the passive detection algorithm can handle

dealing with stative passive, as it cannot be reliably identified using the pattern based automatic means that we employ here.

For detecting sentences in passive voice, we use the following algorithm.

1. *Find all tokens that are located in the left sentence bracket.*
   By definition, only verbal material may occur in the left sentence bracket.

2. *Find all tokens that are located in the right sentence bracket.*
   By definition, only verbal material may occur in the right sentence bracket.

3. *The left sentence bracket is occupied with a form of* werden.

   (a) *The right sentence bracket contains one token with POS tag VVPP.*
   This indicates that the main verb is a form of *werden*-passive. Patterns of the kind listed in the first two lines in table 9.5 can be detected with this condition.

   (b) *The right sentence bracket contains at least one inflected form of* werden, *and a passive participle with POS tag VVPP.*
   This indicates that the sentence is in passive voice. The constraint that a passive participle must be present in addition to a form of *werden* makes sure that active future sentences are not mistaken to be passive.
   Patterns of the kind listed in the third and fourth line in table 9.5 can be detected with this condition.

Verb	ON	OA
entscheiden	Bundessozialgericht/NN	
erklären	Richter/NN	Sauberkeit/NN
gehören	Wäsche/NN	
übergeben	Mitglieder/NN	Volksbegehren/NN
müssen	Ministerium/NN	Gesetzesentwurf/NN
überprüfen	Ministerium/NN	Gesetzesentwurf/NN
verüben		Bombenanschlag/NN
geben	es/PPER	Bekennermeldungen/NN
treten	Familien/NN	
beklagen	sie/PPER	Bedrohung/NN
verüben		Bombenanschlag/NN

Table 9.6: The first few entries in the list of verb argument tuples.

4. *The left sentence bracket is occupied with a form of* sein *or a modal verb, and the right sentence bracket contains at least one inflected form of* werden*, and a passive participle with POS tag VVPP.*
   This indicates that the sentence is in passive voice.

   Patterns of the kind listed in the last two lines in table 9.5 can be detected with this condition.

Note that this filter will not apply to forms of passive with fronted verb complex, such as in ***Bestätigt worden*** *ist der Präsident (The president was confirmed)*.

Extracted data

The result of the extraction procedure described in the previous two sections is a large list of verbs and their arguments. This list contains 15 709 590 verb argument tuples. The first few entries in this list are shown in table 9.6. From this list, all possible verb-subject and verb-object pairs are extracted. This yields two unfiltered lists of 7 315 875 verb-subject pairs and 6 689 417 verb-accusative object pairs.

Freq.	Pair	English
8353	spielen Rolle	*play role*
3188	stellen Frage	*pose question*
2841	lösen Problem	*solve problem*
2810	geben Grund	*be reason*
2515	geben Möglichkeit	*exist possibility*
2462	treffen Entscheidung	*make decision*
2422	verletzen Mensch	*hurt human*
2311	geben Problem	*be problem*
2212	erzählen Geschichte	*tell story*
2184	stellen Antrag	*file request*
1999	töten Mensch	*kill human*
1879	sollen Mark	*shall Mark*
1875	kosten Mark	*cost Mark*
1864	machen Sinn	*make sense*
1862	zahlen Mark	*pay mark*
1810	beantworten Frage	*answer question*
1803	verdienen Geld	*earn money*
1733	führen Gespräch	*conduct conversation*
1656	machen Spaß	*make fun*
1613	erreichen Ziel	*reach goal*

Table 9.7: The twenty highest ranked pairs of verbs and accusative objects in the TüPP-D/Z corpus

9.3.3 Evaluation of extracted verb-object-pairs

Table 9.7 shows the twenty highest ranked verb-object pairs in the TüPP-D/Z treebank. The most frequent pair is *spielen – Rolle (play – role)*, which ocurs 8353 times. The pair-occurrence frequency drops quite quickly, the second pair *stellen – Frage (pose – question)* only occurs less than half as often as *spielen – Rolle*. This is an effect of Zipf's law, which we will return to later. Altogether, the pairs in the list all turn out to be sensible combinations. The only exception to this is the pair *sollen – Mark*. It stems from sentences such as *"das Auto soll 30.000 Mark kosten, sagt der Händler" ("the dealer says that the car shall cost 30.000 Mark")*, where either the parser misannotated the

modal *sollen*, or the pair-extractor did not properly recognize the modal in verb-second position and therefore failed to extract the infinitive *kosten* in the verb complex.

In the form as shown, it is hard to see whether *the sum* of all nouns that occur with a specific verb yields a good description of the corresponding semantic field. Table 9.8 shows all the pairs in the co-occurrence list that involve the verb *essen* and that occur at least 10 times in the corpus. With a few exceptions (we will discuss these below), all nouns are from the semantic field of food. The most frequent pair *essen – Fleisch (eat meat)* occurs 97 times, followed by *essen – Fisch (eat fish)*, which only occurs 64 times. Thus, the co-occurrence counts reflect quite well that *essen* requests types of food in its accusative object argument position.

The top-ranked pairs are fairly prototypical, as we would expect from a pair that occurs in multiple places in a text. However, the frequency ranking can neither be taken as a hint of its level of generality *(eat – cucumbers vs. eat – food)*, nor how typical a food it is that is eaten: In table 9.7, the pair *essen – Banane* is ranked higher than the more general pair *essen – Obst*. Furthermore, bread is a very typical food to be eaten, and the corresponding pair *essen – Brot* occurs 29 times in the corpus. On the other hand, it might be just as typical (at least in Germany) to eat *Schnitzel*, but the pair *essen – Schitzel (eat – cutlet)* is not among the highest ranked pairs. In fact, it occurs only once in the whole corpus. The quality of the list is directly affected by the size of the source corpus: the information about selectional preferences that is contained in the list becomes more and more reliable the more different pairs of verbs and nouns can be extracted. Considering that a relatively typical pair like *essen – Schnitzel* only occurs once, even a very large corpus such as TüPP-D/Z with its 200 million tokens might not be large enough to reflect the real distribution of the pairs.

Furthermore, the lists are subject to Zipf's law, which says that the number of times a pair occurs in the corpus is inversely proportional to its position in the list. We extracted a total of 3 202 927 pairs from the TüPP-D/Z corpus, of which 79% (2 539 954 pairs) occur only once in the whole corpus. The remaining 21%, 662 973 pairs, occur more than once (with *spielen – Rolle* on rank 1, occurring 8353 times). This distribution is illustrated in figure 9.2, which shows a plot of the number of pairs that occur n times. The Y-axis plots on a logarithmic scale the total number of distinct pairs

Freq.	Pair	English
97	essen Fleisch	*eat meat*
64	essen Fisch	*eat fish*
51	essen Ei	*eat egg*
35	**essen Tag**	*eat day*
31	essen Suppe	*eat soup*
30	essen Rindfleisch	*eat beef*
	essen Kuchen	*eat cake*
29	essen Brot	*eat bread*
25	essen Schweinefleisch	*eat pork*
20	essen Banane	*eat banana*
19	essen Schokolade	*eat chocolate*
18	**essen real**	*eat* $real_{adj}$
	essen Gemüse	*eat vegetables*
17	essen Obst	*eat fruit*
16	essen Apfel	*eat apple*
14	essen Pizza	*eat pizza*
	essen Kartoffel	*eat potato*
13	**essen Kind**	*eat child*
12	essen Wurst	*eat sausage*
	essen Tomate	*eat tomato*
	essen Salat	*eat salad*
11	**essen Teller**	*eat plate*
	essen Käse	*eat cheese*
10	essen Würstchen	*eat sausage*
	essen Spaghetti	*eat spaghetti*
	essen Pilz	*eat mushroom*
	essen Mittag	*eat noon*
	essen Mensch	*eat human*
	essen Huhn	*eat chicken*
	essen Geld	*eat money*

Table 9.8: All verb-noun pairs involving the verb *essen* with frequencies ≥ 10.

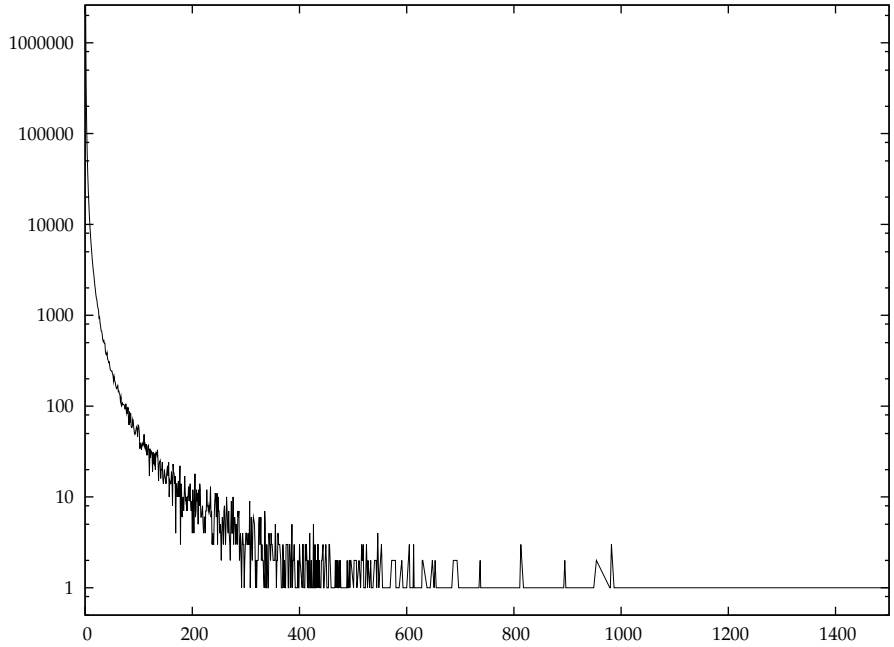


Figure 9.2: Zipfian distribution of verb-object pairs in TüPP-D/Z. The X-axes plots the number of times n that a pair occurs in the corpus (the interval is cut off at $n = 1\,500$). The cumulative number of pairs that each occur n times is plotted along the Y-axis. Note that the Y-axis is scaled logarithmically.

that each occurs n times in the corpus, and the X-axis plots n. For example, there are 4 470 different pairs each of which occurs 13 times in the corpus, i.e. $x = 13$ and $y = 4\,470$. There are 2 539 954 different pairs which only occur once, i.e. $x = 1$ and $y = 2\,539\,954$. The pair *spielen – Rolle* is the only pair that occurs 8 353 times, here $x = 8\,353$, and $y = 1$.

Thus, the frequency of pairs is distributed very unevenly, and furthermore, "good" pairs occur in all ranks in the list – we cannot generally say that the mass of singleton pairs is only "trash", and those pairs that are ranked high are "good". We must keep in mind that the frequency list is just a "collection of special cases" – it contains the counts of the pairs as they occur in *one single concrete corpus*. Although these counts may be a good indication of what the real distribution might be, they just as well might not, as in the cutlet example. We can formalize the effect of selectional preferences by a probability distribution of the appearance of nouns in the ar-

gument positions of the verb. On the basis of this probability distribution, we can then test whether a pair is likely according to that distribution, or not. However, we do not know this distribution, but given our raw data, and some assumptions on the general form of probability distribution that applies to our task, we can *estimate* the missing parameters. That way, we introduce an additional step of abstraction: instead of using a collection of concrete frequency samples from the corpus, we use these pairs to estimate the underlying probability distribution, and based on this distribution, we rank the pairs. This rank gives us a measure of how strong a collocation the pair is, i.e. to which degree the argument is a central member of the required semantic class.

9.4 Log-likelihood ratios

As motivated in the previous section, an additional step of abstraction is necessary to determine which pairs occur together with statistical significance, and which pairs do not. We use the log-likelihood ratio for this task. The log-likelihood ratio was introduced by Dunning (1993) in the context of determining verb-noun pairs. In the following, we will report Dunning's arguments without going into the mathematical details.

Extracting verb-noun pairs is counting words, or more accurately, counting co-occurrences of words. Statistically, this can be formalized as a long series of repeated Bernoulli trials, very similar to tossing a coin. Under the assumption that words are independent of each other and the probability that a word occurs is the same no matter where it occurs, the number of times that a certain word occurs within the next n words is a random variable which is binomially distributed. It is well known that the assumptions of independence and stationarity are actually not true – of course, the probability of the occurrence of a word depends on its location and its context – but they reduce the complexity of the statistical linguistic model to feasible ranges. It has been shown that in spite of these simplifications, the statistical models work well.

Dunning points out that for testing statistical significance, frequently tests are used that assume an underlying normal distribution, since these tests are well known and easy to use. He argues that this works well for binomially distributed trials, since the normal distribution very closely ap-

proximates the binomial distribution, but only if the distribution does *not* involving rare events. However, collocations of words in a text *mostly consist of rare events*. This is directly related to the Zipfian distribution of words in a corpus: A content-bearing word that occurs more than five times can already be considered a frequent word, as we saw in section 9.3.3. Dunning shows that with rare events, the normal distribution is a very bad approximation of the binomial distribution.

As an alternative, he suggests the usage of likelihood ratios, which behave much more stable in the relevant range of occurrence frequencies. The basic idea of likelihood ratios in the context of extracting verb-noun pairs is to compare two different hypotheses H_1 and H_2, where H_1 is the hypothesis that the verb and the noun are *statistically independent*, i.e. they are not a typical pair, and H_2 is the hypothesis that they are *statistically dependent*. The log-likelihood ratio

$$\lambda = \frac{H_1}{H_2}$$

can be interpreted as a measure of the collocational strength of a pair, and therefore can be used as a statistical filter on verb-noun pairs, as described in the following section.

Apart from log-likelihood ratios, we experimented also with Latent Semantic Clustering (Rooth, 1998), which has the advantage over the log-likelihood approach that it can produce smoothed clusters of verb-object pairs and discover additional pairs that do not occur in the source data as such. However, the poor results that we obtained for the log-likelihood ratio and for our following in-depth assessment of the reasons (see section 9.5.2) led to the decision that we would not further consider the LSC variant at this point.

9.5 Experiments

We restricted the following experiments to pairs of verbs and accusative objects. Empirical inspection showed that verb-object clusters exhibit considerably more coherence than verb-subject clusters, which is a vital prerequisite for the acquisition of selectional preferences of satisfying quality.

The list of pairs contained a total of 3 202 927 unique entries. We deleted all pairs that occurred less than three times in the list. Due to the Zipfian

Verb	Objects
essen	speise lebensmittel same hunger getreide kuh besonderes schwein produkt beste zeug tier salz menge rind jahr scheiße nahrung mal kind sache gurke milch gift angst dach geld stück pflanze hund berliner kaffee leut ding kohl deutsche finger wasser seele rest
kauen	kaugummi nagel
einwerfen	scheibe fensterscheibe schaufensterscheibe fenster schaufenster münze brief werbung droge

Table 9.9: Verb-object pairs determined by log-likelihood filtering

Verb	Objects
essen	Gruppe natGegenstand Nahrung Tier Zeit Kommunikation Gefuehl Substanz Relation Koerper Pflanze Besitz Menge Attribut Geschehen Kognition Artefakt Mensch
kauen	Nahrung Koerper Artefakt
einwerfen	Kommunikation Substanz Nahrung Artefakt Form

Table 9.10: Verb-object pairs after mapping to GermaNet unique beginners

distribution of instances in the list, this removes 89.4%, or 2 863 279 of all pairs, leaving 339 648 unique pairs with a frequency of at least 3.

For all pairs in this filtered verb-object frequency list, we computed the log-likelihood ratio. As a further clean-up-step, we removed the pairs which were classified by the log-likelihood test as statistically independent (at a confidence level of $p = 0.05$). Statistical independence in this context indicates that the data does not provide sufficient evidence to ascertain that the pair actually mirrors the selectional preference of the verb. This left us with 207 206 different verb-object pairs. We then determined the set of selectional preferences for each verb by collecting all nouns that occurred with that verb in the list. A few of the resulting sets are shown in table 9.9.

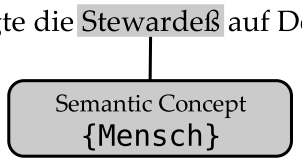


Figure 9.3: The semantic class of a proper noun can be directly determined.

Concept Abstraction The log-likelihood method provides a means to assure that the collocational strength of the verb-object pairs is above chance level, in other words, that the pair is a typical representative of the selectional properties of the verb. However, the sets are still a collection of relatively few but quite specific terms, as obvious from table 9.9. Since later on, the information about selectional preferences is to be applied to *new* data, an additional step of generalization is necessary. To this end, the terms were looked up in GermaNet (Hamp and Feldweg, 1997) and then replaced with the *unique beginner* that dominates the synset the term is a member of.[6] GermaNet contains 22 unique beginners, which are the most general concepts that exist in the network. All unique beginners are listed in table 9.11. The result of this mapping for the three verbs in the above example is shown in table 9.10. It should be noted that in some cases, the mapping of concrete nouns to abstract concepts actually *increases* noise. In table 9.9 the pair *essen – Angst (eat – fear)*, due to a very idiosyncratic use of *eat* in context, is mapped to the abstract concept *Gefühl (feeling)*, thus overly emphasizing this exceptional use. However, our method of determining semantic compatibility as explained in the next section will mostly remain unaffected by this noise, since after intersecting concept sets, most likely only the relevant concepts are retained.

Semantic Compatibility It is fairly straightforward to determine the semantic class of a common noun (i.e. here: a *non-pronoun*), since it has overt semantic content. The determination of the semantic class can be as easy as a direct lookup (for example, in GermaNet). Figure 9.3 illustrates this for

[6] In case of an ambiguous term (i.e. a term that occurs in more than one synset, the term may be mapped to more than one unique beginner.

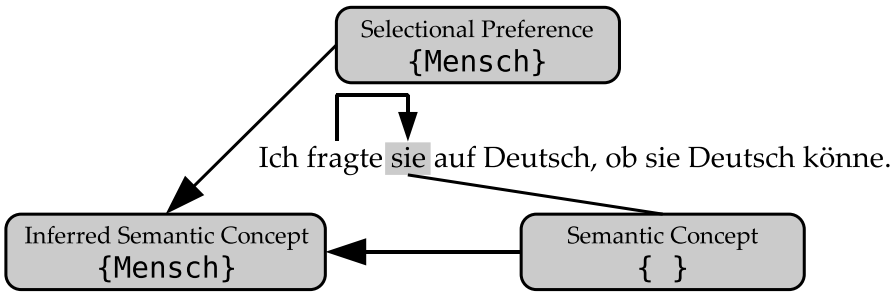


Figure 9.4: The semantic class of a pronoun must be inferred using selectional preferences.

the noun *Stewardeß*,[7] which maps to the GermaNet concept class *Mensch*.

As hinted at before, pronouns are semantically empty by themselves. Their semantic interpretation exclusively depends on the meaning of their antecedent, which, of course, in the context of anaphora resolution, leads to a circular problem. To break this circle, we note that the semantic class of a pronoun in a position that is restricted by the verb's selectional restrictions must obey these restrictions just like a common noun or noun phrase. Thus, the (unknown) semantic class of the pronoun must be a member of the selectional preferences set of the verb. On the other hand, since the pronoun has the same semantic class as its antecedent, the semantic class of the antecedent must be a member of the verb's selectional preferences set as well (see figure 9.4). We can thus establish a relation of *semantic compatibility*: A pronoun and an antecedent are semantically compatible if the intersection of their concept sets (as looked up in GermaNet) is not empty. This is illustrated in the following example:

(7) a. Als dann auch noch **die Stewardeß** eine Bestellung vom
When then also yet the stewardess an order from
Nachbarsitz, "einen Tomatensaft mit Pfeffer und
the neighboring seat, "a tomato juice with pepper and
Salz", nicht verstand, mußte ich es einfach wissen.
salt", not understood, had I it just to know.

'And then when the stewardess did not understand an order from the neighbor seat, "a tomato juice with pepper and salt", I just had to know.'

[7] The TüBa-D/Z corpus that this example is taken from uses old German spelling, which we retain.

natürlicher Gegenstand	Motiv
Gruppe	Nahrung
natürliches Phänomen	Form
Tier	Zeit
Kommunikation	Gefühl
Substanz	Ort
Relation	Körper
Pflanze	Menge
Besitz	Attribut
Geschehen	Kognition
Artefakt	Mensch

Table 9.11: GermaNet's 22 unique beginners

b. Ich fragte **sie** auf Deutsch, ob sie Deutsch könne.
I asked her in German, whether she German could.

'I asked her in German whether she spoke German.'

The personal pronoun *sie*, which is the accusative object of *fragte*, refers to *Stewardeß* in the previous sentence. The synset that contains *Stewardeß* is dominated by the unique beginner *Mensch*. The selectional preferences set of *fragen* contains one element after lookup, which is again *Mensch*. The intersection of both sets is non-empty. Therefore *sie* and *Stewardeß* are taken as semantically compatible.

In the literature on pronoun resolution for English, similar approaches of employing selectional preferences have been suggested by Dagan and Itai (1990), who describe a system that relies on selectional constraints of verbs *only* to determine antecedents. Ge et al. (1998) use selectional preferences as a feature of semantic restriction on potential antecedents in their statistical approach, and Kehler et al. (2004) provide an in-depth study on the utility of predicate-argument frequencies for English pronoun resolution to which we will return later.

9.5.1 Feature representation

We integrated information about selectional preferences into the resolution process as a feature of semantic compatibility, represented by the intersec-

tion of the concept sets of the pronoun and the candidate antecedent. The semantic compatibility feature must have a form usable by TiMBL. We experimented with three variants of representation. The first variant employs one single binary-valued feature, which is 1 if the intersection is non-empty, i.e. the pronoun and the antecedent are semantically compatible. Otherwise, the feature is set to 0. The second variant is a bitvector representation. As mentioned, the concept sets consist of GermaNet unique beginners. The maximum number of elements in a concept set is therefore bound by the number of unique beginners in GermaNet.[8] The intersection can thus be expressed as a vector with one component for each unique beginner, where a component is 1 if the corresponding unique beginner is present in the intersection, and 0 if it is not. The bitvector that is generated for the concept set {`Mensch`} in the above example is illustrated in figure 9.5.

The bitvector representation substantially increases the dimension of the feature space presented to the TiMBL classifier, but provides also more information, both about the cardinality of the intersected concept sets, and the exact concepts that are shared. Since the TiMBL memory-based learner is known for preferring more compact feature representations over more verbose ones, for the third variant, we determined for each candidate how often it occurs with one of the unique beginners. We then included from the intersection only at most three unique beginners that the respective candidate was mapped to most often.

[8] There are 22 unique beginners in GermaNet. For technical reasons, we added the *Tops* concept as a 23^{rd} component to the actual technical representation of the bitvector. The *Tops* concept is an artificial root that turns GermaNet into a connected graph. No synsets are directly dominated by *Tops*, so the corresponding vector component will always be 0.

	Unique beginners	**Bitvector**
1	natürlicher Gegenstand	0
2	Motiv	0
3	Gruppe	0
4	Nahrung	0
5	natürliches Phänomen	0
6	Form	0
7	Tier	0
8	Zeit	0
9	Kommunikation	0
10	Gefühl	0
11	Substanz	0
12	Ort	0
13	Relation	0
14	Körper	0
15	Pflanze	0
16	Menge	0
17	Besitz	0
18	Attribut	0
19	Geschehen	0
20	Kognition	0
21	Artefakt	0
22	Mensch	1

Figure 9.5: Bitvector representation of the intersection of the semantic classes of the pronoun and candidate antecedent in example (7)

	Precision	Recall	F-Measure
Baseline	0.664	0.457	0.541
post-processed	0.736	0.963	0.835
Experiment I (single)	0.663	0.458	0.542
post-processed	0.737	0.963	0.835
Experiment II (bitvector)	0.662	0.459	0.542
post-processed	0.736	0.963	0.834
Experiment III (most frequent)	0.662	0.459	0.542
post-processed	0.736	0.963	0.834

Table 9.12: Results of integrating the semantic class intersection feature. Note the equal results, especially for experiment II and III.

9.5.2 Results and discussion

We carried out three experiments. In each experiment, we added one of the new features to the standard feature set of the hybrid resolution system. We described this standard feature set in section 8.2.4 of chapter 8 (page 180 ff). We did not perform any instance sampling. The performance of the system in this standard configuration served as the baseline for the experiments that follow.

Experiment I used the single binary compatibility feature, in **Experiment II** we integrated the bitvector representation of the intersected concept sets, and finally **Experiment III** included the ranked features. Table 9.12 shows the results - both with and without applying the closest subject postprocessor.

For all experiments, the results remain virtually unchanged in comparison to the baseline. For the non-post-processed experiments, f-measure shows a slight increase due to the small rise of recall. Precision drops by 0.002 for the bitvector representation and for the reduced intersection in experiment III and 0.001 for the single feature representation.

The performance of the system with engaged post-processor shows analog behavior.

The results show that the memory-based classifier cannot benefit from the additional semantic features, neither in the bitvector representation, nor in the more compact representations indicating semantic compatibil-

Fold	Rank													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
0	6	12	4	5	9	7	11	10	14	**13**	8	1	2	3
1	6	12	5	4	9	7	11	10	14	**13**	8	1	2	3
2	6	12	4	5	9	7	11	10	14	**13**	8	1	2	3
3	6	12	4	5	9	7	11	10	14	**13**	8	1	2	3
4	6	12	5	4	9	7	11	10	14	**13**	8	1	2	3
5	6	12	4	5	9	7	10	14	11	**13**	8	1	2	3
6	6	12	5	4	9	7	11	10	**13**	14	8	1	2	3
7	6	12	4	5	9	7	10	11	14	**13**	8	1	2	3
8	6	12	4	5	9	7	14	10	11	**13**	8	1	2	3
9	6	12	5	4	9	7	11	10	14	**13**	8	1	2	3

Table 9.13: Feature ranking for Experiment I which employs the single compatibility feature (number 13, see table 9.14 for the list of feature numbers).

ity. The inspection of the internal ranking that TiMBL computes for each feature in the training phase corroborates this finding. TiMBL determines this ranking using the Gain Ratio measure, which is a weighted variant of Information Gain. Gain Ratio determines independently for each feature its contribution to the knowledge of the target class (see chapter 7).

For Experiment I, the situation is shown in table 9.13. The table shows a separate ranking for each fold, since they differ slightly. The numbers in the "ranking" column refer to the feature numbers listed in table 9.14.[9] The relevant new feature is SEMCOMPATSINGLE and has the number 13 (printed in bold-face). The feature is ranked second to last[10], except in fold 6, where it ranks one place higher. This clearly shows that the new feature does not add any relevant new information to the resolution process.

The feature ranking for Experiment II is shown in table 9.15 (p. 236). Again, the rankings are broken down by fold. The relevant features (numbered 13-35), are printed in bold-face. Unlike Experiment I, where the position of the single compatibility feature can be clearly identified, there is no predominant position of the bitvector features in the ranking, although

[9] Please refer to section 8.2.3 for a description of the other features.

[10] Recall that features 1-3 are only used for bookkeeping in the processing chain and ignored by TiMBL.

Number	Feature	Description
0	ARTNUM	article number
1	PRONID	pronoun markable ID
2	NPID	NP markable ID
3	PRONTYPE	pronoun type
4	PRONGF	grammatical function of the pronoun
5	NPGF	grammatical function of the noun phrase
6	NPTYPE	type of NP
7	DEFINITENESS	type of article
8	EMBEDDING	embedding of NP
9	DIRECTION	direction of relation
10	PARAGF	parallelism of grammatical function
11	SENTDIST	sentence distance
12	WORDDIST	word distance
13	**SEMCOMPATSINGLE**	**semantic compatibility (single feature)**
13–35	**SEMCOMPATBITVEC**	**semantic compatibility (bitvector)**
14/36	GOLDCLASS	gold referential relation

Table 9.14: Features used for the memory-based resolver

most features have a general tendency to always appear in the same region. Feature 21 (which corresponds to the concept *Kommunikation*) appears on rank 2 or 3 in all folds. Feature 20 *(Zeit)* is usually ranked quite high as well, but here, variability is much higher: in folds 0, 2, 5 and 6, feature 20 is on position 4, but in fold 8, it is only on rank 14. This difference of rank is substantial, reasons for which should be expected to be found in the training data. Therefore, we counted how often features 21 and 20 were set to 1 in training samples corresponding to positive and negative pronoun-antecedent pairs, and how often they were 0 (recall that a feature value of 1 means that the corresponding concept occurs in the intersection of the concept sets of the pronoun and the antecedent). Furthermore, we added features 35 and 24 to the benchmark, which correspond to the concepts *Mensch* and *Ort*, respectively. Feature 35 ranks relatively constantly in the medium range between positions 13 and 17, while feature 24 occurs most frequently on one of the last three ranks. Table 9.16 shows the results, again broken down by fold. Columns 4-6 are a four-way count of the cases:

1. Feature value is 1, pair is anaphoric (class *yes*): set/yes

2. Feature value is 0, pair is anaphoric (class *yes*): unset/yes
3. Feature value is 1, pair is not anaphoric (class *no*): set/yes
4. Feature value is 0, pair is not anaphoric (class *no*): set/no

Column 7 contains the ratio of the number of feature vectors where one of the four bitvector components considered is set, and the class is *yes*, to the number of the feature vectors where the class is *no*.

Column 8 finally contains the same ratio for the case where the bitvector components are not set.

In fold 0, we get the following ranking for the four features:

1. Feature 21 *(Kommunikation)*
2. Feature 20 *(Zeit)*
3. Feature 35 *(Mensch)*
4. Feature 24 *(Ort)*

However, as table 9.16 shows, there is no correlation of the feature rank with either the absolute frequency of occurrence nor with the ratio of samples with the feature set and samples with the feature not set. If we order the features according to the number of times they have actually value 1 (which means that the corresponding concept is a member of the intersection of the concept sets of pronoun and antecedent), we get the following:

1. Feature 35 *(Mensch)*
   2238 times (509 times in positive examples, 1729 times in negative examples)
2. Feature 24 *(Ort)*
   889 times (88 times in positive examples, 801 times in negative examples)
3. Feature 21 *(Kommunikation)*
   869 times (125 times in positive examples, 744 times in negative examples)
4. Feature 20 *(Zeit)*
   298 times (13 times in positive examples, 285 times in negative examples)

A ranking by the ratio of the number of times one of the bitvector features is 1 versus the feature is 0 ("set-ratio" in table 9.16) yields

1. Feature 20 *(Zeit)*
2. Feature 24 *(Ort)*
3. Feature 35 *(Mensch)*
4. Feature 21 *(Kommunikation)*

Table 9.16 shows that this pattern remains constant over all folds.

An additional noticeable result as obvious from table 9.15 is that the TiMBL classifier does not recognize the set nature of the additional bitvector features. They are not ranked as a consecutive unity, but instead they are interspersed with the other features. In other words, the classifier treats the bitvector features as *individual semantic properties* of pronouns and antecedents, but not as a measure of semantic compatibility.

These findings lead to the following conclusion. The TiMBL classifier does *not* benefit from a notion of semantic compatibility. This is firstly corroborated by Experiment I, where the single compatibility feature is ranked second to last. Secondly, the feature ranking in Experiment II indicates that the classifier ignores the set character of the intersection as it is represented here in the form of a bitvector. Instead, it treats the individual features separately. This indicates that the additional semantic features were in fact considered by the TiMBL classifier in the sense of the likelihood of some concepts to occur more frequently in referential relations than other. However, the variability of the ranking of these features is quite high, at the same time with a correlation of set/not set frequency and a rank which is quite low, as discussed above. We can therefore conclude that the semantic features in conjunction with the other morphosyntactic features are not informative enough to effectively influence the resolution result.

Fold	Rank																																		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**
0	6	12	**21**	**20**	4	5	**27**	**25**	**17**	**33**	9	**30**	**35**	7	**31**	**28**	**19**	**34**	**23**	**18**	11	10	**22**	**14**	**15**	**32**	36	**13**	**29**	**16**	8	**24**	**26**	1	2	3
1	6	**28**	**21**	12	**20**	5	4	**25**	9	**30**	**33**	**17**	**34**	**27**	**14**	**31**	**35**	**19**	7	**15**	11	**22**	**23**	10	**32**	36	**13**	**18**	**26**	**16**	8	**24**	**29**	1	2	3
2	6	12	**21**	**20**	4	5	**28**	**25**	9	**30**	**33**	**27**	**34**	**35**	**17**	7	**19**	**15**	11	**22**	**23**	10	**31**	**32**	36	**13**	**18**	**14**	**26**	**29**	**16**	8	**24**	1	2	3
3	6	12	**21**	4	5	**25**	**28**	**20**	9	**17**	**27**	**30**	**33**	7	**34**	**35**	**32**	**22**	**19**	**31**	11	**23**	10	**15**	36	**13**	**18**	**16**	**29**	**14**	8	**24**	**26**	1	2	3
4	6	**21**	12	5	4	**20**	**25**	9	**15**	**19**	**28**	**33**	**27**	**35**	**17**	7	**30**	**34**	**31**	**23**	11	**22**	10	36	**32**	**18**	**13**	**14**	**16**	**26**	8	**24**	**29**	1	2	3
5	6	12	**21**	**20**	4	5	**25**	**28**	**30**	**34**	9	**33**	**27**	**17**	7	**35**	**31**	**22**	**23**	10	**15**	**19**	**32**	**14**	11	**13**	36	**18**	**26**	**29**	8	**16**	**24**	1	2	3
6	6	**21**	12	**20**	5	4	**28**	**17**	9	**25**	7	**30**	**33**	**27**	**35**	**34**	**19**	11	36	**31**	10	**32**	**22**	**23**	**15**	**13**	**18**	**26**	**14**	8	**24**	**16**	**29**	1	2	3
7	6	**21**	12	**27**	4	5	**28**	9	**30**	**25**	**20**	**17**	**19**	**33**	**35**	7	**29**	**31**	**34**	**15**	**32**	**22**	10	11	**23**	36	**13**	**14**	**18**	**26**	**24**	**16**	8	1	2	3
8	6	**21**	12	**17**	**27**	4	5	**25**	**33**	9	**30**	**28**	**35**	**20**	**34**	**19**	**31**	7	**22**	**32**	10	11	**18**	**23**	**15**	**13**	36	**14**	**24**	**26**	**16**	**29**	8	1	2	3
9	6	**21**	12	5	4	**20**	**25**	**27**	9	**30**	**17**	**33**	**28**	7	**19**	**35**	**34**	**31**	11	10	**32**	**15**	36	**22**	**23**	**13**	**24**	**18**	**29**	**26**	**14**	8	**16**	1	2	3

Table 9.15: Feature ranking for Experiment II which employs the bitvector representation of semantic compatibility (numbers 13-35, see table 9.14 for the list of feature numbers).

Fold	Feature	Set		Unset		Set-Ratio	Unset-Ratio
		Yes	No	Yes	No		
0	21	125	744	27479	117805	1:5.95	1:4.29
	20	13	285	27591	118264	1:21.92	1:4.29
	35	509	1729	27095	116820	1:3.40	1:4.31
	24	88	801	27516	117748	1:9.10	1:4.28
1	21	118	744	27896	117395	1:6.31	1:4.21
	20	12	284	28002	117855	1:23.67	1:4.21
	35	510	1700	27504	116439	1:3.33	1:4.23
	24	94	784	27920	117355	1:8.34	1:4.20
2	21	119	734	27682	117618	1:6.17	1:4.25
	20	13	261	27788	118091	1:20.08	1:4.25
	35	509	1749	27292	116603	1:3.44	1:4.27
	24	94	797	27707	117555	1:8.48	1:4.24
3	21	113	735	27749	117556	1:6.50	1:4.24
	20	14	270	27848	118021	1:19.29	1:4.24
	35	498	1700	27364	116591	1:3.41	1:4.26
	24	92	779	27770	117512	1:8.47	1:4.23
4	21	121	724	27564	117744	1:5.98	1:4.27
	20	14	290	27671	118178	1:20.71	1:4.27
	35	498	1672	27187	116796	1:3.36	1:4.30
	24	89	750	27596	117718	1:8.43	1:4.27
5	21	116	748	27585	117704	1:6.45	1:4.27
	20	14	281	27687	118171	1:20.07	1:4.27
	35	482	1678	27219	116774	1:3.48	1:4.29
	24	82	762	27619	117690	1:9.29	1:4.26
6	21	119	725	26678	118631	1:6.09	1:4.45
	20	13	262	26784	119094	1:20.15	1:4.45
	35	505	1693	26292	117663	1:3.35	1:4.48
	24	92	756	26705	118600	1:8.22	1:4.44
7	21	120	739	27320	117974	1:6.16	1:4.32
	20	10	265	27430	118448	1:26.50	1:4.32
	35	495	1722	26945	116991	1:3.48	1:4.34
	24	89	764	27351	117949	1:8.58	1:4.31
8	21	109	702	28066	117276	1:6.44	1:4.18
	20	10	265	28165	117713	1:26.50	1:4.18
	35	495	1706	27680	116272	1:3.45	1:4.20
	24	81	722	28094	117256	1:8.91	1:4.17
9	21	128	740	27282	118003	1:5.78	1:4.33
	20	13	291	27397	118452	1:22.38	1:4.32
	35	512	1715	26898	117028	1:3.35	1:4.35
	24	90	816	27320	117927	1:9.07	1:4.32

Table 9.16: Bitvector value distribution per class

9.6 Evaluation

At first sight, the experimental results in the previous section seem to reject the hypothesis that additional information based on semantic features can improve the performance of our machine-learning-based anaphora resolution system. However, anaphora is, and will always be, a linguistic phenomenon that is most strongly influenced by meaning, in the sense of coherence, as characterized in chapter 1. The reasons for the little effect of adding the new semantic features are therefore not to be sought in the concept of their integration in machine-learning-based approaches as such, but rather in a combination of the specific properties of the data employed in this research - both of the text to be analyzed and of the source for semantic information. We will examine the system under three aspects that are of relevance in order for new information to be effective. They are:

- **Applicability**
- **Coverage**
- **Discriminativeness**

9.6.1 Applicability

In order to be beneficial for a machine learning system, a feature must describe a phenomenon in the data to be processed that is actually present with sufficient frequency. If this is not the case, the effect of the feature in context of the other features and in the overall evaluation will be quite limited even though, in isolation, the feature may be a "good feature", i.e. provide a reliable representation of the phenomenon it is supposed to describe. In other words, the feature must be *applicable* to the data.

We examine in this chapter the influence of a new semantic feature based on selectional preferences that represents the semantic compatibility between a pronoun and a candidate antecedent. As explained, we determined these selectional preferences by extracting all pairs of verbs and their accusative objects from the TüPP-D/Z corpus. We had to restrict our research to verb-object pairs since the variability of the verb-subject pairs that we extracted from the corpus turned out to be too high to deliver reliable selectional preferences, and arguments of other grammatical functions

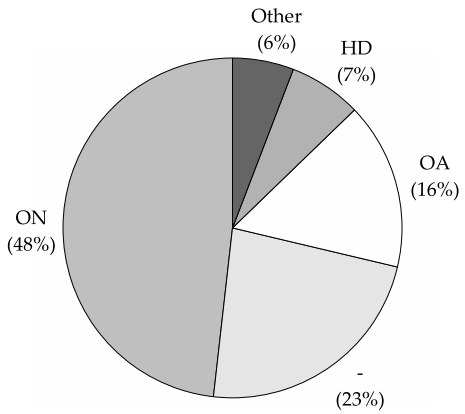


Figure 9.6: Distribution of pronouns in TüBa-D/Z by grammatical function that are in an *anaphoric* relation.

were either too infrequent or not annotated (i.e. parsed) with sufficient accuracy.

As a consequence, the mechanism of determining the semantic compatibility of a pronoun and a potential antecedent (and, on the basis of this, accepting or ruling out a candidate), can of course *only* be applied to pronouns that are accusative objects of the corresponding verb: To pronouns in other locations, the new semantic feature is *not applicable* – it does not properly describe the relevant data.

Figure 9.6 illustrates the distribution of the grammatical function of those pronouns in the TüBa-D/Z treebank that were (manually) classified to be in an *anaphoric* relationship to some antecedent. The treebank contains 13 824 of such pronouns. Of these, 6 664 are subjects (grammatical function ON, 48%), 3 193 are in a position with no grammatical function assigned (23%), 960 have the grammatical function HD (7%), and 807 (6%) are assigned various other grammatical functions (some of which due to annotation errors). But most importantly, there are 2 200 pronouns (16%) that occur with a grammatical function of OA, the accusative object. While this is a substantial amount, we nevertheless note that the applicability of the semantic compatibility feature is limited to only 16% of all relevant instances.

In order to compute the semantic compatibility feature for the pronouns in OA position, the selectional preferences of the verb that the pronoun is

Verb-object pairs		
total	6 689 417	
unique	3 202 927	100%
$n \geq 3$	339 648	11%
$n < 3$	2 863 279	89%
significant (log-likelihood)	207 206	6%

Table 9.17: Lengths of verb-object pair lists after application of several filters.

an object of must be known. We therefore determined the number of pronouns that occur with verbs with known selectional preferences and found 1 748 pronouns. The set of pronouns which are in an anaphoric relation to which the semantic compatibility feature can be applied is thus only 13% of the total 13 824 pronouns. The subset of these pronouns in the baseline setting without postfiltering reaches a precision of 78.8%, recall of 68% and f-measure of 0.73 - i.e. only using morphosyntactic features the performance on this class of pronouns is far superior than the performance on all pronouns (precision: 64.4%, recall: 42.8%, f-measure: 52.1%).

The applicability of the semantic compatibility feature is thus limited due to the small fraction of the data that the feature can be applied to at all, and, independently of that, the exceptionally high performance of the resolver on this fraction. It is obvious that it would require extremely strong features to visibly affect the total performance of the resolution approach under these circumstances.

9.6.2 Coverage

The second aspect, coverage, concerns the question whether enough data is available to get access to the required information. Our task is especially affected by issues of coverage in three points: Firstly, the extraction of verb-object pairs from the TüPP-D/Z corpus, secondly, the lookup of concrete nouns in GermaNet for determining an abstract concept set, and finally, the assignment of concept sets to pronouns (as explained in section 9.5).

As described earlier, the automatically annotated TüPP-D/Z treebank is a very large corpus, containing the enormous amount of 194 826 942 tokens in the version annotated with grammatical functions. From this corpus,

we extracted a total amount of 6 689 417 verb-object pairs. This is the total number of pairs, including pairs that occur multiple times. The number of unique pairs is about half this size and comprises 3 202 927 pairs. In section 9.3.3, we pointed out that like most collections of data of this kind, the list has a Zipfian distribution, i.e. the majority of all pairs occurs seldom. We decided to apply a threshold of a frequency of at least 3 for a pair to be used in the further process (which is still quite low), as pairs below would add too much noise to the selectional preferences extracted. This means that 2 863 279 (89%) of the pairs are removed from the list, leaving only 339 648 (11%). Using the χ^2 distributed function -2λ of the log-likelihood values λ, we removed the insignificant pairs from this list (according to the χ^2 test), leaving a further reduced list of 207 206 pairs, which is only about 6% of the original size (see table 9.17).

Thus, what seems as an enormous amount of data in the beginning is quickly reduced to a rather small set, once it is cleaned up such that the amount of potential noise in the data has been reduced to a reliable degree.

We explained how verb-object-pairs are extracted from the TüPP-D/Z corpus and combined into selectional preferences sets on a per-verb basis. Since these sets contain very concrete nouns which are likely not applicable to new data, the additional generalization step of mapping all nouns to GermaNet unique beginners is necessary. As with any manually created resource, the number of concepts in GermaNet is naturally limited. We identified this mapping step as a further potential source of problems due to coverage. GermaNet version 5.11, the version that we based our work on, contains a total of 53 312 synsets with 76 519 lexical units. The noun subclass, which is the one relevant for this task, comprises 38 725 synsets with 55 755 lexical units. The 1 188 first articles of the TüBa-D/Z treebank that we used for our experiments contain 172 977 markables, of which 141 091 are common and proper nouns. We determined how many of these could be mapped to GermaNet unique beginners and found that for 109 617 markables (78%), corresponding synsets could be found in GermaNet. This amount is satisfactory, and we state that the concept mapping to GermaNet does not constitute a major coverage problem.

As explained, since pronouns are semantically empty by themselves, we depend on the verb that the pronoun is the accusative object of to determine the concept set - using the selectional preferences of the verb. It is

therefore vital that the verbs are all assigned selectional preferences. In the data, there are 3 584 verbs that occur with pronouns in accusative object position. However, only 519 of these could be assigned selectional preferences sets, that is about 15%. This is problematic, since without the assignment of selectional preferences to the argument position, the semantic concept of the pronoun cannot be determined. In practice, the situation is eased by the fact that there is a Zipfian distribution on the verbs as well. Although selectional preferences can only be determined for 519 unique verbs, a large number of pronouns actually occurs with one of these verbs, so more pronouns are assigned concept sets than initially expected. Of the 2 200 pronouns in OA position, 1 748 (79%) have concept sets. However, even though this coverage problem is mild by itself, the number of pronouns that the semantic features can actually be applied to is further reduced. Here, coverage has a direct effect on applicability, as the set of pronouns that the new features are applicable to amounts to only 13% of the total number of pronouns.

9.6.3 Discriminativeness

The final aspect, discriminativeness, concerns the question whether a feature reliably partitions the data into complementary classes. The semantic compatibility feature is based on the hypothesis that the majority of correct antecedents are semantically compatible to the pronoun, while the majority of those NPs that are not antecedents are semantically incompatible. If this hypothesis does not hold true, the potential of the classifier to discriminate between (potentially) correct antecedents and most likely non-antecedents on the basis of this becomes quite limited – in other words, during the training phrase the classifier would recognize that the feature's influence on the actual classes is small and rank it down.

In the TüBa-D/Z data, we assessed for all 2 200 pronouns in OA position the number of markables that were annotated as their antecedents, the fraction of these which were semantically compatible, the semantically incompatible ones, and finally the antecedents that were not assigned concept sets. Furthermore we determined the same values for the markables within a window of three sentences around the pronoun that were *not* antecedents. The results are summarized in table 9.18.

Of the 16 241 antecedents of pronouns in OA position, 4 157 are deter-

Antecedents	16 241
compatible	4 157
incompatible	1 248
no concept set	10 836
***Non*-Antecedents**	143 323
compatible	71 211
incompatible	21 270
no concept set	50 842

Table 9.18: Distribution of antecedents and non-antecedents of pronouns in OA position with respect to their semantic compatibility.

mined as semantically compatible (i.e. the intersection of the antecedent's concept set and the pronoun's selectional preferences set is non-empty). 1 248 are semantically incompatible due to their empty intersection of a non-empty concept set of the antecedent and a non-empty selectional preferences set of the pronoun. Finally, 10 836 antecedents have no concept sets assigned, so their semantic compatibility cannot be computed. Of these 10 836 antecedents, only 626 are common nouns with POS tag NN, while the others are pronouns of different kinds and antecedents with other parts of speech, such as cardinal numbers. Some are also annotation errors. Thus, for 67% of all antecedents of pronouns in OA position, the semantic feature cannot apply, since it was not possible to compute a concept set, leaving only 33% to which the feature would be applicable. But even for this subset it shows that the hypothesis that an antecedent is also semantically compatible only holds true in 77% of the cases.

There are 143 323 non-antecedents in the three-sentence windows around the pronouns in OA position, 71 211 are semantically compatible, and 21 270 are semantically incompatible. Thus, in the fraction of non-antecedents for which it was possible to determine a concept set, there are 3.35 times as many semantically compatible NPs than semantically incompatible NPs. The ratio of compatible and incompatible concept sets for NPs that actually are antecedents is 3.33:1. Thus, the ratio of compatible and incompatible concepts is constant irrespective of whether an NP is an antecedent of a pronoun or not. The distribution of semantic compatibility of antecedents and non-antecedents cannot be taken as a strong basis for

discriminating between the two classes.

At this point it should be noted that the three aspects of applicability, coverage, and discriminativeness obviously overlap, and cannot be considered in separation. The fact that no concept set can be assigned to more than half of all antecedents is both an issue of coverage, but also an issue of applicability, as concept sets cannot be determined at all for these types of markables. This finally affects discriminativeness, as for both antecedents and non-antecedents more than half of the relevant cases get lost.

9.6.4 Conclusion

In this chapter, we presented experiments that integrated a new semantic feature based on selectional preferences of verbs on their accusative objects into the anaphora resolution process. The results show that the classifier could not benefit from the new feature, neither in a compact representation of semantic compatibility using one binary feature, nor in a bitvector representation. The performance remains virtually unchanged. Thus, our findings for German are in line with what Kehler et al. (2004) report for English. They implemented two resolution systems, one based on Maximum Entropy Modeling, and the other using a Naive Bayes approach. Both systems used a set of morphological, positional, syntactic, and lexical features at their core. The resolution decisions were postfiltered using predicate-argument statistics acquired from the Topic Detection and Tracking (TDT-2) corpus. These selectional preferences were based on 1 321 072 subject-verb pairs, 1 167 189 verb-object pairs, and 301 477 possessive noun pairs. Kehler et al. do not mention in their paper whether these numbers are unique pairs or total pairs. They found that the MaxEnt postfilter only improved performance by marginal 0.5%. They conclude that

- predicate-argument statistics are of little predictive power to a pronoun interpretation system,
- plain morphosyntactic features suffice to successfully resolve most of the pronouns,
- selectional preferences provide a poor substitute for world knowledge.

Our experiments described in this chapter were based on 6 689 417 total verb-object pairs, yielding 3 202 927 unique pairs. Thus in any case,

we used substantially greater amounts of data. Nevertheless, our results corroborate Kehler et al.'s findings, and vice versa. Furthermore, our in-depth study shows that one important reason for the non-utility lies in the characteristic of the data itself: Reliable selectional preferences can only be extracted from reliable argument positions, which restricts the domain the preferences are applicable to to a limited subset. The low variability in these argument positions stems from the fact that they underlie relatively strong syntactic restrictions. But of course, if a pronoun occurs in this position, these restrictions apply as well, which is why rather simple morphosyntactic features are already sufficient to correctly resolve most of these pronouns. This is what we referred to as an issue of applicability.

Given that two studies for two different languages and very different amounts of data yield essentially exactly the same results, the conclusion to be drawn is that the use of selectional preferences for pronoun resolution *in addition* to other morphosyntactic features will not improve the performance of a resolution system. Matters are different when selectional preferences are used as the only information source, such as in the work by Dagan and Itai (1990). Here the performance reached is comparable to the performance of systems using "lightweight" morphosyntactic features. However, this hardly justifies the engineering overhead for acquiring selectional preferences on a large scale, which is substantially higher than that for extracting morphosyntactic features.

This leads to the question of how to go on. The type of pronouns that occur most frequently by far (at least in our corpus), are pronouns in subject position. We explained that subject-verb relations exhibited too much variability – at least for the given amount of data, which, after proper filtering, turned out to be not as much as it might have seemed at first sight. Thus, an extremely specialized very large scale system for reliably recognizing only subjects and verbs within a sentence could help gather very large amounts of subject-verb pairs as the basis for selectional preferences. Next, more sophisticated methods of semantic class abstraction than we applied in our experiments could be applied to these pairs. Here, the methods of Abe and Li (1996) or combined methods such as the one developed by Wagner (2005) seem useful. Further, recent methods of acquiring semantic relatedness from the web, such as the work by Ponzetto and Strube (2007) or Zesch et al. (2008) can help improve and extend semantic concept sets

built from selectional preferences by incorporating related concepts gathered from other sources.

The pronoun type that leaves most ample room for improvement are attributive possessive pronouns. They are in a position that is most unrestricted - neither by the noun they modify nor by a verb. Still, the most likely referent for a possessive pronouns is the one that is the most salient. Thus, a pronoun resolution system that is *integrated* with a sophisticated high precision/high recall full NP resolver that is capable of generating full referential chains and that models the real time dynamics of discourse salience might be able to acquire the necessary information to improve the resolution accuracy of pronouns in such weakly restricted positions. Such a more global view of pronoun resolution could be combined with ranking systems for anaphora as recently suggested by Denis and Baldridge (2007), who implement a Maximum Entropy based resolution system that imposes a ranking on *all* potential antecedents at once, and who achieve promising improvements.

While pronoun resolution approaches based on binary classification seem exhausted, the combination of approaches that are capable of more accurately modeling the nature of competition of potential antecedents in discourse with additional global data leaves promising options for future improvement.

Chapter 10

Conclusion

This dissertation was concerned with the resolution of anaphora. Anaphora is a semantic relation that expresses the identity of reference between a pronoun and another noun phrase. The purpose of anaphora is to establish cohesion between the utterances in a discourse. As such, anaphora is an important contributor to meaning.

We implemented and discussed two approaches to anaphora resolution which we both applied to the TüBa-D/Z treebank of German newspaper text.

Rule-based anaphora resolution The first approach is a rule-based system, a re-implementation of Lappin and Leass' Resolution of Anaphora Procedure for German (RAP-G). At the core of the system, there is a module to determine the salience of all entities in the discourse. The candidate with the highest salience is selected as the antecedent of a pronoun. RAP-G models the dynamic change of salience: The salience of a candidate increases if it refers to an entity which is mentioned frequently in the discourse. The salience decreases the further apart from the pronoun the candidate is located. RAP-G implements syntactic filters and rules which allow it to check whether a pair of a pronoun and a potential antecedent that occur within the same sentence satisfy the binding principles. Although the system was initially designed as a baseline for comparing rule-based and machine-learning-based approaches, it provided interesting results of its own about the influence of language-dependent word-order and text genre on the importance of the factors contributing to salience.

A hybrid approach to anaphora resolution The second approach is a hybrid system, which combines rule-based pre- and postfilters with a memory-based main resolution component. We found that the performance of the system is competitive to that of classic rule-based systems. The hybrid architecture combines the strengths of rule-based and data-driven approaches within one system: High-precision linguistic rules are employed in areas where this is feasible – in limited domains such as morphological filtering or heuristic post-selection of antecedents, which can be specified using sets of linguistic rules of manageable size and complexity. For the resolution task proper, which would require complex data and rules, the memory-based resolution module is employed.

Instance sampling In the training data for the resolution module, negative (non-anaphoric) training samples outweigh the positive samples (anaphoric samples) by a factor of more than four. This skewed distribution leads to a bias of the classifier towards negative classification. We used instance sampling to adjust the ratio of positive and negative samples in a controlled way with the aim of reducing the classifier's bias. We compared several methods that randomly select pairs, sampling methods that remove samples based on linguistic considerations such as the position relative to a correct antecedent, methods that remove samples based on their expected potential of partitioning the sample space, and finally methods that employ incremental learning, i.e. methods that only learn *difficult* samples. We found that the reduction of samples works best when the remaining samples stem from all areas of the original sample space, a structure created by random instance sampling. We further found that sampling methods based on incremental learning are effective when the training data is to be optimized for size while minimizing performance sacrifices.

Semantic features We finally returned to the semantic nature of anaphora and incorporated semantics as a feature of semantic compatibility between a pronoun and a potential antecedent into the resolution process. Assuming that the set of potential semantic classes of a pronoun is equal to the set of selectional preferences that a verb puts on its objects, we computed a feature of semantic compatibility in terms of the intersection of the semantic concept sets of a pronoun and its potential antecedent. The results show

that the classifier can not benefit from the new feature. To research and explain the reasons, we performed a detailed analysis of the data and the system. The outcome of these studies reveals a data structure that leaves the domain that selectional restrictions can be applied to fairly limited. Our study shows that in this domain, traditional morphosyntactic features are sufficient for reaching state-of-the-art performance. Furthermore, even though we extracted selectional preferences from very large amounts of data using our specifically designed distributed software environment, after properly filtering the data, only a fairly small number of pairs remained usable.

Future research We identify two areas of future research. The first one pertains to the specific properties of our data, while the second one concerns more general considerations.

Pronouns in subject position. With respect to the first area, we saw that pronouns in subject position cover almost half of all pronouns. As mentioned, the new semantic compatibility feature cannot be applied to pronouns in subject position since selectional restrictions on the subject position turned out to be too weak. Here, a specialized system for detecting subjects with high reliability might help to annotate additional resources on a very large scale, thus multiplying the amount of available data and thereby improving reliability. More sophisticated approaches of generalizing semantic concepts, such as proposed by Wagner (2005) seem useful. Further, recent methods of acquiring semantic relatedness from the web, such as the work by Ponzetto and Strube (2007) or Zesch et al. (2008) can help improve and extend semantic concept sets built from selectional preferences by incorporating related concepts gathered from other sources. Furthermore, the semantic knowledge can be extended using reliable sources such as the SALSA corpus (Burchardt et al., 2006). In the light of the results obtained for accusative objects, the performance improvement may not be too large, however, even smaller improvements might have higher impact due to the large number of subject pronouns in the data.

Attributive possessive pronouns. The resolver performs worst on attributive possessive pronouns, since they occur in a both syntactically as well as semantically relatively unrestricted environment. Approaches exploiting

verb frames such as selectional preferences or SALSA cannot be used here. Instead, discourse-global properties could help to add additional information for performance improvements. This leads us to the more general, second area of further research.

Integrated discourse-global models of resolution. The work by Yang et al. (2003) and Denis and Baldridge (2007), among others, shows that the performance on the anaphora resolution task can be improved by adopting a more global view on a discourse. The competition models that the authors describe in their work are a first step in this direction. However, even in these approaches, the task of resolving pronouns is still handled largely separate from coreference resolution of full NPs. A resolution model that is capable of formulating both tasks as an *integrated* process seems to have potential to take the performance in the field some steps further.

Appendix A

STTS – The Stuttgart Tübingen Tagset

This appendix lists the Stuttgart Tübingen Tagset (STTS, Schiller et al. 1999) as used in the TüBa-D/Z treebank. This table is a verbatim copy of the one given in the treebank's stylebook (Telljohann et al., 2006).

POS tag	Description	Examples
ADJA	attributive adjective	*[das] große [Haus]*
ADJD	adverbial or predicative adjective	*[er fährt] schnell, [er ist] schnell*
ADV	adverb	*schon, bald, doch*
APPR	preposition; left circumposition	*in [der Stadt], ohne [mich]*
APPRART	preposition + article	*im [Haus], zur [Sache]*
APPO	postposition	*[ihm] zufolge, [der Sache] wegen*
APZR	right circumposition	*[von jetzt] an*
ART	definite or indefinite article	*der, die, das, ein, eine*
CARD	cardinal number	*zwei [Männer], [im Jahre] 1994*
FM	foreign language material	*[Er hat das mit "] A big fish [" übersetzt]*
ITJ	interjection	*mhm, ach, tja*
KOUI	subordinating conjunction with *zu* + infinitive	*um [zu leben], anstatt [zu fragen]*
KOUS	subordinating conjunction with clause	*weil, daß, damit, wenn, ob*

POS tag	Description	Examples
KON	coordinative conjunction	*und, oder, aber*
KOKOM	particle of comparison, no clause	*als, wie*
NN	noun	*Tisch, Herr, [das] Reisen*
NE	proper noun	*Hans, Hamburg, HSV*
PDS	substituting demonstrative pronoun	*dieser, jener*
PDAT	attributive demonstrative pronoun	*jener [Mensch]*
PIS	substituting indefinite pronoun	*keiner, viele, man, niemand*
PIAT	attributive indefinite pronoun without determiner	*kein [Mensch], irgendein [Glas]*
PIDAT	attributive indefinite pronoun with determiner	*[ein] wenig [Wasser], [die] beiden [Brüder]*
PPER	irreflexive personal pronoun	*ich, er, ihm, mich, dir*
PPOSS	substituting possessive pronoun	*meins, deiner*
PPOSAT	attributive possessive pronoun	*mein [Buch], deine [Mutter]*
	relative pronoun	
PRELS	substituting	*[der Hund,] der*
PRELAT	attributive	*[der Mann ,] dessen [Hund]*
PRF	reflexive personal pronoun	*sich, einander, dich, mir*
PWS	substituting interrogative pronoun	*wer, was*
PWAT	attributive interrogative pronoun	*welche [Farbe], wessen [Hut]*
PWAV	adverbial interrogative or relative pronoun	*warum, wo, wann, worüber, wobei*
PROP	pronominal adverb	*dafür, dabei, deswegen, trotzdem*
PTKZU	*zu* + infinitive	*zu [gehen]*
PTKNEG	negation particle	*nicht*
PTKVZ	separated verb particle	*[er kommt] an, [er fährt] rad*
PTKANT	answer particle	*ja, nein, danke, bitte*
PTKA	particle with adjective	*am [schönsten], zu [schnell]*

POS tag	Description	Examples
	or adverb	
TRUNC	truncated word - first part	*An– [und Abreise]*
VVFIN	finite main verb	*[du] gehst, [wir] kommen [an]*
VVIMP	imperative, main verb	*komm [!]*
VVINF	infinitive, main	*gehen, ankommen*
VVIZU	infinitive + *zu*, main	*anzukommen, loszulassen*
VVPP	past participle, main	*gegangen, angekommen*
VAFIN	finite verb, aux	*[du] bist, [wir] werden*
VAIMP	imperative, aux	*sei [ruhig !]*
VAINF	infinitive, aux	*werden, sein*
VAPP	past participle, aux	*gewesen*
VMFIN	finite verb, modal	*dürfen*
VMINF	infinitive, modal	*wollen*
VMPP	past participle, modal	*[er hat] gekonnt*
XY	non-word containing special characters	*D2XW3, letters*
$,	comma	**,**
$.	sentence-final punctuation	**. ? ! ; :**
$(	other sentence internal punctuation	**- [ ] ( )**

Appendix B

Morphological Feature Combinations in STTS

This appendix lists the valid combinations of morphological features and POS tags in the Stuttgart Tübingen Tagset (STTS, Schiller et al. 1999) as used in the TüBa-D/Z treebank. This table is a verbatim copy of the one given in the treebank's stylebook (Telljohann et al., 2006).

POS	feature combination	comments
ADJA	case number gender	underspecified for gender if plural noun is underspecified, e.g. *die/np* nordhessischen/np* Grünen/np** invariant local description e.g. *Berliner/**** cardinal numbers as abbreviation: full morphology e.g. *im 4./dsn Jahrhundert/dsn*
APPR	case	without case if a prepositions takes another PP as complement, e.g. *bis/_ zu/d einer/dsf Woche/dsf*
APPRART	case number, gender	
APPO	case	
ART	case number gender	
NE	case number gender	

NN	case number gender	underspecified for gender, e.g. *Abgeordnete* (in plural), *Leute*
PDAT	case number gender	
PDS	case number gender	
PIAT	case number gender	plural is underspecified for gender, e.g. *lauter/****, see also 'PIS or PIAT' below
PIDAT	case number gender	*solch/**** (cf. *manch, welch, all*), see also 'PIS or PIDAT' below
PIS	case number gender	underspecified: *man/ns** *nichts/**** (cf. *nix, sowas*) PIS or PIAT: *allerhand/**** (cf. *allerlei, allzuviel, dergleichen, derlei, etwas, genausoviel, genug, genügend, keinerlei, mehr, reichlich, soviel, viel, wenig, weniger, zuviel, zuwenig*) PIDAT or PIS: *sowas/**** (cf. *paar, bißchen*)
PPER	case number gender person	
PPOSAT	case number gender	
PPOSS	case number gender	
PRELAT	case number gender	
PRELS	case number gender	plural is underspecified for gender
PRF	case number gender person	*sich*: underspecified for gender
PWAT	case number gender	plural is underspecified for gender *wessen/****
PWS	case number gender	underspecified for gender: plural forms and *wer, wem, wen*
VAFIN	person number mood tense	
VAIMP	person number	

VMFIN	person number mood tense	
VVFIN	person number mood tense	
VVIMP	number	German has only second person imperative forms

Values of morphological features

Feature	Values
case	n (nominative), g (genitive), d (dative), a (accusative), * (underspecified)
gender	m (masculine), f (feminine), n (neuter), * (underspecified)
number	s (singular), p (plural), * (underspecified)
mood	i (indicative), k (subjunctive; German 'Konjunktiv')
person	1 (first), 2 (second), 3 (third), * (underspecified)
tense	s (present), t (past)

Appendix C

Category Labels

This appendix lists all possible category node labels as defined in the TüBa-D/Z annotation scheme. This table is a verbatim copy of the one given in the treebank's stylebook (Telljohann et al., 2006).

Node Labels	Description
Phrase Node Labels	
ADJX	adjectival phrase
ADVX	adverbial phrase
DP	determiner phrase (e.g. *gar keine*)
FX	foreign language phrase
NX	noun phrase
PX	prepositional phrase
VXFIN	finite verb phrase
VXINF	non-finite verb phrase
Topological Field Node Labels	
LV	resumptive construction (Linksversetzung)
C	complementizer field (C-Feld)
FKOORD	coordination consisting of conjuncts of fields
KOORD	field for coordinating particles
LK	left sentence bracket (Linke (Satz-)Klammer)
MF	middle field (Mittelfeld)
MFE	middle field between VCE and VC
NF	final field (Nachfeld)

PARORD	field for non-coordinating particles
VC	verb complex (Verbkomplex)
VCE	verb complex with the split finite verb of *Ersatzinfinitiv* constructions
VF	initial field (Vorfeld)
FKONJ	conjunct consisting of more than one field
Root Node Labels	
DM	discourse marker
P-SIMPX	paratactic construction of simplex clauses
R-SIMPX	relative clause
SIMPX	simplex clause

Appendix D

Edge Labels

This appendix lists all possible edge labels as defined in the TüBa-D/Z annotation scheme. This table is a verbatim copy of the one given in the treebank's stylebook (Telljohann et al., 2006).

Edge Labels	Description
Edge Labels denoting Heads and Conjuncts	
HD	head
-	non-head
KONJ	conjunct
Complement Edge Labels	
ON	nominative object (i.e. subject; also clausal subjects)
OD	dative object
OA	accusative object
OG	genitive object
OS	sentential object
OPP	prepositional object
OADVP	adverbial object
OADJP	adjectival object
PRED	predicate
OV	verbal object
FOPP	facultative (i.e. optional) prepositional object, passivized subject (*von*-phrase)
VPT	separable verb prefix

APP	apposition
Modifier Edge Labels	
MOD ON-MOD, OA-MOD, OD-MOD, OG-MOD, OPP-MOD, OS-MOD, PRED-MOD, FOPP-MOD, OADJP-MOD, V-MOD, MOD-MOD	ambiguous modifier modifiers modifying complements or modifiers e.g. V-MOD = modifier of the verb
Edge Labels in Split Coordinations	
ONK, ODK, OAK, FOPPK, OADVPK, PREDK, MODK, V-MODK	second conjunct (K) in split coordinations e.g. ONK = second conjunct of a nominative object
Edge Label denoting Structural Expletive	
ES	Vorfeld-*es*
Secondary Edge Labels	
REFVC REFMOD REFINT REFCONTR	dependency relation between: two verbal objects in VC two ambiguous modifiers a phrase internal part and its modifier control verb and its complement across clause boundaries

Appendix E

Named Entity Categories and Edge Labels

Named entities are either annotated with the category node EN-ADD, or with a secondary edge labeled `EN`. For a detailed discussion of annotation rules for named entities, see section 5.1.3 and the TüBa-D/Z stylebook (Telljohann et al., 2006).

Labels	Description
Phrase Node Labels	
EN-ADD	proper noun or named entity (additional label)
Secondary Edge Label	
EN	phrase internal relation between two parts of a proper noun

Bibliography

Abe, N. and Li, H. (1996). Learning word association norms using tree cut pair models. In *Proceedings of the 13^{t}h International Conference on Machine Learning*, Bari, Italy.

Aha, D., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6:37–66.

Aït-Mokhtar, S., Chanod, J.-P., and Roux, C. (2002). Robustness beyond shallowness: incremental deep parsing. *Nat. Lang. Eng.*, 8(3):121–144.

Alshawi, H. (1987). *Memory and Context for Language Interpretation*. Cambridge University Press.

Aone, C. and Bennett, S. (1995). Evaluating automatic and manual acquisition of anaphora categories. In *Proceedings of the 31st Annual Meeting of the ACL (ACL '95)*, pages 122–129, Cambridge, MA, USA.

Azzam, S. (1996). Resolving anaphors in embedded sentences. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 263–268, Morristown, NJ, USA. Association for Computational Linguistics.

Baldwin, B. (1997). Cogniac: high precision coreference with limited knowledge and linguistic resources. In *Proceedings of the ACL/EACL '97 workshop on Operational Factors in Practical, Robust Anaphora Resolution*, pages 38–45, Madrid, Spain.

Beaver, D. I. (2004). The optimization of discourse anaphora. *Linguistics and Philosophy*, 27(1):3–56.

Bikel, D. M., Schwartz, R., and Weischedel, R. M. (1999). An algorithm that learns what's in a name. In *Machine Learning*, volume 34, pages 211–231.

Black, E., Abney, S., Flickenger, D., Gdaniec, C., Grishman, R., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J., Liberman, M., Marcus, M., Roukos, S., Santorini, B., and Strzalkowski, T. (1991). A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proceedings of Fourth DARPA Speech and Natural Language Workshop*, pages 306–311.

Boyd, A., Gegg-Harrison, W., and Byron, D. (2005). Identifying non-referential *it*: a machine learning approach incorporating linguistically motivated patterns. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP*, pages 40–47, Ann Arbor. Association for Computational Linguistics.

Brennan, S. E., Friedman, M. W., and Pollard, C. J. (1987). A centering approach to pronouns. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, pages 155–162, Morristown, NJ, USA. Association for Computational Linguistics.

Büring, D. (2005). *Binding Theory*. Cambridge University Press.

Briscoe, E. and Carroll, J. (1993). Generalised probabilistic lr parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19(1):25–60.

Bußmann, H. (2002). *Lexikon der Sprachwissenschaft*. Kröner, Stuttgart, third edition.

Burchardt, A., Erk, K., Frank, A., Kowalski, A., Padó, S., and Pinkal, M. (2006). The SALSA Corpus: a German Corpus Resource for Lexical Semantics. In *Proceedings of LREC 2006*, Genoa, Italy.

Chomsky, N. (1993). *Lectures on Government and Binding – The Pisa Lectures*. Mouton de Gruyter, Berlin and New York, 7th edition.

Church, K. (1988). A stochastic parts program and noun phraser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143.

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

Cost, S. and Salzberg, S. (1993). A weighted nearest neighbour algorithm for learning with symbolic features. *Machine Learning*, 10:57–78.

Daelemans, W. and van den Bosch, A. (2005). *Memory-based Language Processing*. Cambridge University Press.

Daelemans, W., Zavrel, J., van der Sloot, K., and van den Bosch, A. (2005). TiMBL: Tilburg memory based learner– version 5.1–reference guide. Technical Report ILK 01-04, Induction of Linguistic Knowledge, Computational Linguistics, Tilburg University.

Dagan, I. and Itai, A. (1990). Automatic processing of large corpora for the resolution of anaphora references. In *Proceedings of the 13th conference on Computational linguistics*, pages 330–332, Morristown, NJ, USA. Association for Computational Linguistics.

Denis, P. and Baldridge, J. (2007). A Ranking Approach to Pronoun Resolution. In *Proceedings of IJCAI 2007*, pages 1588–1593.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.*, 19(1):61–74.

Eisenberg, P. (1999). *Grundriß der deutschen Grammatik – Band 2: Der Satz*. J.B. Metzler, Stuttgart, Weimar.

Evans, R. (2001). Applying Machine Learning Toward an Automatic Classification of *It*. *Literary and Linguistic Computing*, 16(1):45–57.

Filippova, K. (2005). A memory-based learning approach to pronominal anaphora resolution in german newspaper texts. Master's thesis, University of Tübingen, Tübingen, Germany.

Ge, N., Hale, J., and Charniak, E. (1998). A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–170.

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

Grosz, B. J. and Sidner, C. L. (1986). Attentions, intentions and the structure of discourse. *Computational Linguistics*, 12:175–204.

Halliday, M. and Hasan, R. (2006). *Cohesion in English*. Longman.

Hamp, B. and Feldweg, H. (1997). Germanet – a lexical-semantic net for german. In *Proceedings of the ACL/EACL-97 Workshop on Automatic Information Extraction and Building of Semantic Resources for NLP Applications*, pages 9–15, Madrid.

Höhle, T. N. (1986). Der Begriff 'Mittelfeld'. Anmerkungen über die Theorie der topologischen Felder. In Schöne, A., editor, *Textlinguistik contra Stilistik. Akten des VII. Internationalen Germanisten-Kongresses Göttingen 1985*, pages 329–340. Niemeyer, Tübingen.

Hinrichs, E. W., Filippova, K., and Wunsch, H. (2005a). A Data-driven Approach to Pronominal Anaphora Resolution in German. In Angelova, G., Bontcheva, K., Mitkov, R., Nicolov, N., and Nicolov, N., editors, *Proceedings of the 5th International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*, pages 239–245, Borovets, Bulgaria.

Hinrichs, E. W., Filippova, K., and Wunsch, H. (2005b). What Treebanks Can Do For You: Rule-based and Machine-learning Approaches to Anaphora Resolution in German. In Civit, M., Kübler, S., and Martí, M. A., editors, *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, pages 77–88, Barcelona, Spain.

Hinrichs, E. W. and Wunsch, H. (2009). Selectional preferences for anaphora resolution. In Nerbonne, J. and Hinrichs, E. W., editors, *Theory and Evidence in Semantics*. CSLI Publishers.

Hirschmann, L. and Chinchor, N. (1997). *MUC-7 Coreference Task Definition*. Version 3.0.

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

Jackendoff, R. S. (1972). *Semantic Interpretation in Generative Grammar*. The MIT Press.

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449.

Kamp, H. (1981). A Theory of Truth and Semantic Representation. In Janssen, T., Groenendijk, J., and Stokhof, M., editors, *Formal Methods in the Study of Language*. Mathematisch Centrum, Amsterdam.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic*. Kluwer Academic Publishers.

Kehler, A. (1997). Probabilistic coreference in information extraction. In *In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 163–173.

Kehler, A., Appelt, D., Taylor, L., and Simma, A. (2004). The (non)utility of predicate-argument frequencies for pronoun interpretation. In Susan Dumais, D. M. and Roukos, S., editors, *HLT-NAACL 2004: Main Proceedings*, pages 289–296, Boston, Massachusetts, USA. Association for Computational Linguistics.

Kennedy, C. and Boguraev, B. (1996). Anaphora for Everyone: Pronominal Anaphora Resolution without a Parser. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96*, pages 113–118.

Lappin, S. and Leass, H. J. (1994). An algorithm for pronominal anaphora resolution. *Comput. Linguist.*, 20(4):535–561.

Lasnik, H. (1989). *Essays on Anaphora*. Kluwer Academic Publishers.

Lees, R. and Klima, E. S. (1969). Rules for English Pronominalization. In Reibel, D. A. and Schane, S. A., editors, *Modern Studies in English – Readings in Transformational Grammar*, pages 145–159. Prentice-Hall, Englewood Cliffs NJ.

Levelt, W. J. M. (1991). *Speaking – from intention to articulation*. MIT Press, Cambridge, MA.

McCarthy, J. F. and Lehnert, W. G. (1995). Using decision trees for coreference resolution. In *In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1050–1055.

Miller, G. A., Fellbaum, C., Kegl, J., and Miller, K. J. (1988). Wordnet: an electronic lexical reference system based on theories of lexical memory. *Revue quebecoise de linguistique*, 17(2):181 – 213.

Mitkov, R. (1998). Robust pronoun resolution with limited knowledge. In *ACL-36: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Lin-*

guistics, pages 869–875, Morristown, NJ, USA. Association for Computational Linguistics.

Mitkov, R. (2002). *Anaphora Resolution*. Pearson Education, Edinburgh and London.

Müller, C. (2006). Automatic detection of nonreferential it in spoken multiparty dialog. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, Italy. The Association for Computer Linguistics.

Müller, F. H. (2004a). Annotating Grammatical Functions in German Using Finite-State Cascades. In *Proceedings of COLING 2004*, Geneva, Switzerland.

Müller, F. H. (2004b). Stylebook for the Tübingen Partially Parsed Corpus of Written German (TüPP-D/Z).

Müller, F. H. (2007). *A Finite-State Approach to Shallow Parsing and Grammatical Functions Annotation of German*. PhD thesis, Eberhard Karls Universität Tübingen, Tübingen, Germany.

Müller, M.-C. (2008). *Fully Automatic Resolution of It, This and That in Unrestricted Multi-Party Dialog*. PhD thesis, University of Tübingen, Tübingen, Germany.

MUC-6 (1995). Coreference task definition (v 2.3, 8 sep 95). In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 335–344.

MUC-7 (1997). Coreference task definition (v 3.0, 13 jul 97). In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.

Müller, C., Rapp, S., and Strube, M. (2001). Applying co-training to reference resolution. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 352–359, Morristown, NJ, USA. Association for Computational Linguistics.

Naumann, K. (2006). Manual for the annotation of in-document referential relations. Technical report, Seminar für Sprachwissenschaft, Abteilung Computerlinguistik, Universität Tübingen.

Ng, V. (2005). Machine learning for coreference resolution: from local classification to global ranking. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 157–164, Morristown, NJ, USA. Association for Computational Linguistics.

Ng, V. (2007). Semantic class induction and coreference resolution. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 536–543, Prague, Czech Republic.

Ng, V. and Cardie, C. (2001). Improving machine learning approaches to coreference resolution. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 104–111, Morristown, NJ, USA. Association for Computational Linguistics.

Ng, V. and Cardie, C. (2002). Identifying anaphoric and non-anaphoric noun phrass to improve coreference resolution. In *Proceedings of the 19^{t}h International Conference on Computational Linguistics (COLING-2002)*.

Paice, C. and Husk, G. (1987). Towards the automatic recognition of anaphoric features in english text: text impersonal pronoun "it". *Computer Speech and Language*, 2:109–132.

Poesio, M. (2000). Coreference. In Mengel, A., Dybkjaer, L., Garrido, J., Heid, U., Klein, M., Pirrelli, V., M. Poesio, M., Quazza, S., Schiffrin, A., and Soria, C., editors, *MATE Dialogue Annotation Guidelines*, chapter 2.4. `http://www.ims.uni-stuttgart.de/projekte/mate/mdag/`. Institut für maschinelle Sprachverarbeitung, University of Stuttgart.

Poesio, M. (2004). The MATE/GNOME Proposals for Anaphoric Annotation, Revisited. In *Proceedings of SIGDIAL*, Boston.

Pollard, C. and Sag, I. A. (1992). Anaphors in english and the scope of binding theory. *Linguistic Inquiry*, 23(2):261–303.

Pollard, C. and Sag, I. A. (1994). *Head-Driven Phrase Grammar*. University of Chicago Press.

Ponzetto, S. P. and Strube, M. (2006). Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the main conference on Human Language Technology Conference of the North*

American Chapter of the Association of Computational Linguistics, pages 192–199, Morristown, NJ, USA. Association for Computational Linguistics.

Ponzetto, S. P. and Strube, M. (2007). Deriving a Large Scale Taxonomy from Wikipedia. In *Proceedings of AAAI 2007*, pages 1440–1445.

Postal, P. M. (1966). On so-called "pronouns" in english. In *19th Monograph on Language and Linguistics*. Georgetown University Press, Washington, D.C.

Preiss, J. (2002). Anaphora resolution with memory based learning. In *Proceedings of CLUK5*, pages 1–9.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, CA.

Reinhart, T. (1976). *The Syntactic Domain of Anaphora*. PhD thesis, MIT, Cambridge.

Reinhart, T. (1983). *Anaphora and Semantic Interpretation*. Croom Helm, London & Canberra.

Resnik, P. S. (1993). *Selection and Information: A Class-based Approach to Lexical Relationships*. PhD thesis, University of Pennsylviania.

Rooth, M. (1998). Two-dimensional clusters in grammatical relations. In Rooth, M., Riezler, S., Prescher, D., Schulte im Walde, S., Carroll, G., and Beil, F., editors, *Inducing Lexicons with the EM Algorithm*, volume 4 of *AIMS*, pages 7–24. Universität Stuttgart.

Schiehlen, M. (2004). Optimizing Algorithms for Pronoun Resolution. In *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva.

Schiller, A. (1995). Dmor: Benutzeranleitung. Technical report, Universität Stuttgart, Insitut für maschinelle Sprachverarbeitung.

Schiller, A., Teufel, S., Stöckert, C., and Thielen, C. (1999). Guidelines für das Tagging deutscher Textcorpora mit dem STTS. Technical report, IMS Stuttgart / SfS Tübingen.

Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science*, 237:1317–1323.

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

Strube, M. (1998). Never look back: An alternativ to centering. In *Proceedings of the 17th Internation Conference on Computational Linguistics and 36th Annual Meeting of ACL*, pages 1251–1257.

Strube, M. and Hahn, U. (1999). Functional centering – grounding referential coherence in information structure. *Computational Linguistics*, 25(3):309–344.

Strube, M., Rapp, S., and Müller, C. (2002). The influence of minimum edit distance on reference resolution. In *in Proceedings of Empirical Methods in Natural Language Processing Conference*, pages 312–319.

Stuckardt, R. (2001). Design and enhanced evaluation of a robust anaphor resolution algorithm. *Computational Linguistics*, 27(4):479–506.

Stuckardt, R. (2002). Machine-Learning-Based vs. Manually Designed Approaches to Anaphor Resolution: the Best of Two Worlds. In *Proceedings of DAARC 2002*, pages 211–216, Lisbon.

Telljohann, H., Hinrichs, E., Kübler, S., and Zinsmeister, H. (2006). Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Technical report, Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen.

Trushkina, J. (2004). *Morpho-syntactic annotation and dependency parsing of German*. PhD thesis, University of Tübingen, Tübingen, Germany.

van Deemter, K. and Kibble, R. (2001). On coreferring: Coreference in muc and related annotation schemes. *Computational Linguistics*.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). A model-theoretic coreference scoring scheme. In *MUC6 '95: Proceedings of the 6th conference on Message understanding*, pages 45–52, Morristown, NJ, USA. Association for Computational Linguistics.

von Stechow, A. and Sternefeld, W. (1988). *Bausteine syntaktischen Wissens*. Westdeutscher Verlag, Opladen.

Wagner, A. (2005). *Learning Thematic Role Relations for Lexical Semantic Nets*. PhD thesis, University of Tübingen, Tübingen, Germany.

Weischedel, R. M. and Brunstein, A. (2005). *BBN pronoun coreference and entity type corpus*. Linguistic Data Consortium.

Wilkins, W. (1988). Thematic Structure and Reflexivization. In *Syntax and Semantics*, volume 21 *(Thematic Relations)*, pages 191–213. Academic Press, San Diego.

Yang, X., Su, J., and Tan, C. L. (2005). Improving pronoun resolution using statistics-based semantic compatibility information. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 165–172, Morristown, NJ, USA. Association for Computational Linguistics.

Yang, X., Zhou, G., Su, J., and Tan, C. L. (2003). Coreference resolution using competition learning approach. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 176–183, Morristown, NJ, USA. Association for Computational Linguistics.

Zesch, T., Müller, C., and Gurevych, I. (2008). Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC), electronic proceedings*.

Zhao, S. and Ng, H. T. (2007). Identification and resolution of Chinese zero pronouns: A machine learning approach. In *Proceedings of EMNLP-CoNLL*, Prague, Czech Republic.