Universität Hamburg
MIN-Fakultät
Department Informatik

Hybrid Methods of Natural Language Analysis

Doctoral Thesis

submitted by
Kilian A. Foth
of Hamburg

Hamburg, 2006

Diese Arbeit erscheint im Jahr 2007
bei der Shaker Verlag GmbH, Aachen.

Genehmigt von der MIN-Fakultät Department Informatik
der Universität Hamburg auf Antrag von

Prof. Dr.-Ing. Wolfgang Menzel (Erstgutachter)
Department Informatik
Universität Hamburg

Prof. Dr. Christopher Habel (Zweitgutachter)
Department Informatik
Universität Hamburg

Prof. Dr. Joakim Nivre (Externer Gutachter)
School of Mathematics and Systems Engineering
Växjö University

Hamburg, den 25.10.2006

Prof. Dr. Winfried Lamersdorf, Leiter des Department Informatik

In memoriam Peter J. Foth

Contents

Chapter 3

A Dependency Model of German

An important consideration for judging a parser of a natural language is its intended *coverage*. Parsers for computer languages are expected to deal with every instance of input that is allowed according to the language specification; implementations that cover only a subset of a computer language are comparatively rare and are generally considered inferior programs. But with natural languages, the situation is quite different: the task is generally agreed to be so difficult that even parsers for a restricted version can be useful, either for immediate use or as a step on the road to an eventual full analyzer. Furthermore, natural languages rarely have a formal specification; even when various linguistic theories attempt to give a complete account of a language, these accounts rarely attain the same universal acceptance among native speakers as the language itself.

In general, it requires more effort to write a parser the broader its coverage should be. This is obvious for some formalisms; for instance, a generative context-free grammar must contain a production rule for every sequence of constituents that should be allowed as the expansion of another constituent. Thus, the simplest kind of context-free grammar requires separate rules to cover each of the sentences in the following sequence:

- I met him.
- I met him at the bar.
- I met him at the bar on Friday.
- I met him at the bar on Friday at noon.
- I met him at the bar on Friday at noon with trepidation.
- ...

The difficulty for the grammar writer resides not just in having to write more rules as coverage is extended. Rules for covering the example sentences are actually rather easy to invent, since they all follow the same pattern. In fact, an experienced grammar writer will probably be soon subsume all of these rules under a new and more general rule if the formalism allows this, or else choose (or invent) a more powerful one that does.

The main difficulty is rather that of maintaining a reasonable precision while extending the coverage of a formal description. The more production rules a parser has to consider, the more likely it is that they will interact in ways that were not foreseen by the author, and allow alternative interpretations of sentences that were expected to be unambiguous. For instance, all of the prepositions in the example sentences are intended to be part of the verb phrase "I met him"; but a grammar with any reasonable coverage must also contain productions that allow prepositions to be interpreted as part of the preceding noun phrase. It is clear that this possibility leads to many more interpretations, all of which are partially wrong.

It is possible to contain some of this ambiguity by analysing individual phrases more thoroughly. For instance, a grammar might specify that weekdays such as 'Friday' cannot be modified by phrases whose meaning corresponds to a time unit smaller than a day; this would allow phrases such as "a Friday in May" but forbid the phrase "Friday at noon". This, however, requires many times more effort than was necessary to allow the interpretation in the first place.

It might be assumed that a broad coverage would be easier to achieve in an eliminative formalism like WCDG. After all, to allow more constructions, a grammar writer merely has to forbid fewer of them, i.e. write fewer constraints. In fact, simple grammars written just as a proof of concept can consist of very few constraints; for instance Schröder (1995) defines just 14, Schröder (1996) only 21 constraints. But again, the problem lies not so much in covering more expressions, but in ensuring that they are analysed in the intuitively correct way. Therefore, constraint dependency grammars also tend to require more constraints as their coverage extends; alternatively, the constraints become more complicated (since in theory, any two unary or binary constraints could be combined into one simply by connecting their formulas with a logical `AND`). The grammar used in Menzel and Schröder (1999) uses 220 constraints to describe subordinations on 9 different levels, while the grammar for the restricted domain of Verbmobil dialogues described by Foth (1999b) already contains 491 constraints.

This thesis uses a substantially larger dependency grammar of German (Foth, 2004b) than the ones cited above, which contains over 1,000 constraints. Its aims are correspondingly more ambitious:

1. It aims explicitly to cover modern German *entirely* — that is, it should assign a plausible structure to German text without being tuned to a particular text

type or level of speech. Instead, all syntactic structures that can be expected to recur with some regularity should be recognized.

2. It should describe the input adequately enough that a semantic interpretation can in principle be derived from its output. For instance, rather than produce merely structural information, it should make explicit distinctions such as subjects vs. object or complement vs. modifiers, by marking them with different labels.

3. It should be suitably robust, that is, it should notice if an utterance contains grammatical errors, but do not reject it entirely.

3.1 Dependency corpora

The grammar has been successfully applied to the analysis of different text types. A varied corpus of German sentences has been collected that can be used both to test the performance of the grammar and to extract data for statistical experiments. All of these sentences have been annotated with dependency structure as well as edge labels and morpho-syntactic features.

The newspaper text type is represented by dependency versions of the NEGRA (Brants et al., 1997b) and TIGER (Brants et al., 2002) corpora, translated from the original corpora with the tool DEPSY (Daum et al., 2004). These corpora together comprise 61,000 sentences drawn from the archives of German daily newspapers and cover a large range of topics.

A larger corpus has been acquired from the archives of the technical news service `heiseticker.de`.[1] These online newscasts are characterized by a narrower range of topics, predominantly computer-related, a more hurried style with many grammatical errors and slips, and frequent technical jargon. About 263,000 sentences of news items from five years have been automatically annotated with dependency structure by running the transformational algorithm on them repeatedly. The first 50,000 of these analyses were subsequently reviewed manually and freed of parsing errors; these analyses can now be regarded as correct according to our model of German, while the remaining part still contain misanalyses. Frequent experimentation suggests that the automatically parsed part of the corpus exhibits a structural correctness of over 90%.

As an example of formal language, two instances of law texts were fully analyzed. The 2002 version of the constitution of Federal Germany and the German version of the proposed constitution of the European Union were analysed. These texts

[1] All of these newscasts are publicly available; for instance, item number 10,000 can be accessed online at `www.heise.de/newsticker/meldung/10000`.

frequently contain very long sentences, stilted formal phrasing, and complex constructions intended to be interpreted carefully by experts, rather than immediately understood by laypeople. Together these texts contain 3,700 sentences.

The complement to these difficult texts, as it were, is constituted by 20,000 sentences drawn from trivial literature: although published in book form, they are clearly written with immediate understanding in mind. These sentences are on the whole much shorter and easier to analyse, but nevertheless valuable in their own right because they contain constructions which do not occur in any other part of the corpus, such as vocatives or dialogue structures for direct speech.

To represent transcripts of spoken language, one series of German dialogues from the Verbmobil appointment scheduling corpus was selected. Trivial utterances such as single words, noise and isolated repetitions were discarded; the remaining text contains 1,300 sentences. They are rather repetitive in both form and content.

3.2 Specification

3.2.1 Word definition

German texts are analysed sentence by sentence, with each token corresponding to one node in the dependency tree. An automatic tokenizer of German is maintained in the CDG distribution that is accurate to about one error in 20,000 tokens. Punctuation marks are considered individual tokens, and so are generally included when sentence lengths are given, but always form null dependencies rather than being part of the tree proper.

One of the tasks of the parser is to resolve *categorial* and *morpho-syntactic* lexical ambiguity: each word must be classified not only by its syntactic function, but also by its class as defined by the standard classification STTS (Schiller et al., 1999). Both the basic category classification established by the small tag set and the further morpho-syntactic distinctions specified by the large tag set are implemented. Each word in the lexicon bears the attribute `cat`, whose value is one of the 53 tags described by the STTS. Words which can fall into more than one of these categories are represented by multiple lexicon entries.

Words with the appropriate class also bear further features such as `number`, `case` and `gender` which distinguish inflectional variants. The parser must choose a single one[2] of these to represent each ambiguous word form. For instance, the common word 'die' is subject to three orthogonal dimensions of ambiguity (of category, case, and

[2]If the parser can prove that two or more variants will not be distinctive, i.e. any two analyses that differ only in this lexical variation will exhibit exactly the same set of constraint violations, analysis may actually remain ambiguous with respect to them (Foth, 1999a). However, this is a mere optimization to reduce running time; the parser could trivially be modified so that it always selects one of the equivalent alternatives before returning its result.

number), and consequently has 12 different lexicon items. The extreme case of this is constituted by German adjectives; these obey four independent morphosyntactic dimensions and thus theoretically have 72 different feature combinations, which however are represented by just 5 surface forms. Experimentation has shown that no parsing accuracy is lost when some of these forms are merged into partially underspecified lexicon entries, and therefore the version of the lexicon used normally contains just 15 entries for each adjective instead of the maximal 72.

On the other hand, lexical ambiguity based on *semantic* differences is usually not represented. For instance, although the noun 'Schein' can have more than one distinct meaning[3], such variants do not correspond to multiple lexicon entries. Since the parser does not itself create a semantic interpretation, it would usually have no basis on which to prefer one of these variants over the others, therefore such variants would merely introduce additional ambiguity into the parsing problem without contributing to accuracy in any way. However, variants that give rise to syntactic alternation are always treated as different words; for instance, there are separate entries for the masculine and neuter variants of the noun 'Erbe' ('heir'/'inheritance').

3.2.2 Levels of analysis

Typically, non-punctuation words enter a syntactic dependency relation with exactly one other word, except for the finite verb of the main clause, which forms the root of the tree. Null dependencies can be established for more than one proper word in a sentence when it consists of fragments that cannot be analysed as belonging to the same structure; however, this is avoided even to the point of preferring non-projective to fragmented trees. For instance, in the sentence

"Die Klage gegen eBay hatte Randall Stoner eingereicht, ein Fan der Musikgruppe Grateful Dead."
(The suit against eBay had been brought by Randall Stoner, a fan of the group Grateful Dead*.)*
(heiseticker, item 13060)

the right extraposition "ein Fan..." is analysed as an apposition to the name "Stoner" rather than a fragment, even though this crosses the dependency relation between the parts of the auxiliary group.

However, some even more complicated extrapositions do lead to tree fragments:

"Dafür darf sie sogar ihre Maske zerreißen: einer der wenigen pathetischen Momente des Stücks."
(This even allows her to tear her mask: one of the few solemn moments of the piece.)
(TIGER, sentence 35455 (shortened))

[3]It is likely that the different meanings of 'glow', 'appearance', 'certificate', and 'money bill' are genetically related in the sense that each one developed from the previous one; nevertheless, in a synchronic setting they should be considered distinct.

The extraposition "Momente" cannot reasonably be analysed as a paraphrase of any of the previous noun phrases and therefore forms an additional tree root.

As well as a syntactic dependency, an extrasyntactic dependency can be established for each word. Currently, this option is only used in order to connect relative pronouns with their antecedents in the matrix clause. The extrasyntactic level is designated as `REF` because it only contains such reference relations, while the syntactic level is designated `SYN`.

3.2.3 Dependency labels

The syntactic level associates each dependency with one of 35 different labels in order to make finer distinctions than the subordination structure can express. An overview of the labels is given in Table 3.1. They can be roughly classified into the following groups:

1. Relations between function words that are really only morphosyntactic markers and normal content words. The label `KONJ` connects a subordinating conjunction with its associated final or infinite verb (where the verb is the regent). The label `PART` subordinates a circumposition with its corresponding preposition, while the label `PN` subordinates the kernel of a prepositional phrase to the preposition.

2. Relations internal to verb phrases. The label `AUX` connects the parts of a multi-word auxiliary expression, such as "könnte entstanden sein" (might have occurred). This is also the case if the words are distant from each other, as they usually are in German main clauses. The label `AVZ` connects a separable verb prefix to its finite verb.

3. Relations internal to noun phrases. The label `APP` connects the components both of multi-word expressions such as "Karl Meier" and of true appositions such as "Karl Meier, der Rennfahrer" (Karl Meier, the racing driver). The labels `DET` and `ATTR` are used for prenominal elements; they distinguish determining elements such as articles from mere attributes, usually adjectives. `GMOD` connects a genitive nominal modifier to another nominal element.

4. Verb arguments. The labels `ETH`, `EXPL`, `PRED`, `SUBJ`, and `SUBJC`, as well as all variants of the `OBJ?` label, connect elements that complement verbs to the corresponding verb. The true objects are distinguished by their grammatical category (preposition, clause, infinitive, or nominal), and the nominal objects are further distinguished by their syntactic case. Note that both subjects and subject clauses are considered arguments of the verb in the same way as objects.

5. Adverbial modification. The labels `PP` and `ADV` designate prepositional and other adverbial modification (called this although it does not necessarily occur

Label	Function	Example
`<empty>`	used for punctuation	Hallo **!**
`ADV`	adverbial modification	**Wie** geht es dir?
`APP`	apposition	Artikel **3**
`ATTR`	prenominal attribute	der **goldene** Hahn
`AUX`	auxiliary phrases	Du hast **gewonnen**.
`AVZ`	split verb prefixes	Lenken Sie nicht **ab**!
`CJ`	co-ordinated element	Obst und **Gemüse**
`DET`	determiner	**der** goldene Hahn
`ETH`	ethical dative	Daß **mir** das nicht wieder vorkommt!
`EXPL`	expletive pronoun	**Es** ist schlimm, was passiert ist.
`GMOD`	possessive modification	der Name der **Rose**
`GRAD`	nominal degree expression	zehn **Jahre** alt
`KOM`	comparison	weiß **wie** Schnee
`KON`	co-ordinating conjunction	Obst **und** Gemüse
`KONJ`	subordinating conjunction	**Wenn** es regnet, gießt es.
`NEB`	modal subclause	Wenn es **regnet**, gießt es.
`NP2`	stranded NP in co-ordination	Der Wanderer liebt die schöne Müllerin und die **Müllerin** den Jäger.
`OBJA`	direct object	Öffnet das **Tor**!
`OBJA2`	second direct object	Nennt mich **Ismael**.
`OBJD`	indirect object	Öffnet dem **König**!
`OBJG`	genitive object	Wir sind nicht dieser **Meinung**.
`OBJC`	clausal object	Und er sah, daß es gut **war**.
`OBJI`	infinitive object	Es scheint nicht zu **funktionieren**.
`OBJP`	prepositional object	Achte **auf** den Zug!
`PAR`	parenthetic matrix clause	“Ich glaube”, **sagte** er, “das genügt.”
`PART`	discontinuous morphemes	Er lief auf den Turm **zu**.
`PN`	PP kernel	Spring durch den **Reifen**!
`PP`	prepositional modification	Spring **durch** den Reifen!
`PRED`	predicate	Das war erst der **Anfang**.
`REL`	relative clause	Hunde, die **bellen**, beißen nicht.
`S`	main clause	Not **kennt** kein Gebot.
`SUBJ`	surface nominal subject	**Not** kennt kein Gebot.
`SUBJC`	subject clause	Wer die Wahl **hat**, hat die Qual.
`VOK`	vocative	**Herr**, es ist Zeit.
`ZEIT`	nominal time expression	Er wurde **1960** geboren.

Table 3.1: Dependency labels employed in the grammar of German.

with verbs). The rarer labels `ZEIT` and `GRAD` are used for nominal elements with a temporal or measuring sense respectively.

6. Co-ordination. The label `KON` is used both for asyndetic co-ordination and for subordinating a co-ordinating conjunction to the left conjunct. The label `CJ` connects the right conjunct to the conjunction, while the label `NP2` can be used in elliptical co-ordinations to connect a supernumerary or *stranded* nominal element to the conjunction. The label `KOM` is used instead of `KON` for comparing rather than sense-neutral co-ordinations (those formed with words of the class `KOKOM`).

7. Subordination of entire sentences. `NEB` connects a normal subclause to its matrix clause; at both ends the finite verb is supposed to be the attachment point. `REL` is used instead of `NEB` for relative clauses. The label `S` is used for the null dependency at the top of a main clause, but also for a subordinated clause in verb-second position (it could therefore also be classified among the verb arguments). The label `PAR` is used for the inverse relation, when an embedded matrix clause is subordinated *under* its object clause for structural reasons.

8. Textual macrostructure. When the intended hearer of a sentence is explicitly named, the label `VOK` subordinates their name to the sentence itself. (This relation is not really part of syntactic structure; for instance, it does not obey the rules about German topological structure.) Finally, the empty string is used for the null subordination of punctuation marks.

A more detailed discussion of the differences between these labels is given in the annotation guide (Foth, 2004a). It should be understood that the precise label set is not the issue here; a somewhat larger or smaller set could be achieved with trivial modifications of the grammar. For instance, the labels `AUX` and `AVZ`, both of which form verb phrases, occur with disjunct sets of dependents. Therefore they could be unified to a label such as `VP` without loss of expressivity, by replacing the expressions

```
X.label = AUX  and  X.label = AVZ
```

with

```
X.label = VP & X@cat != PTKVZ  and  X.label = VP & X@cat = PTKVZ
```

in all constraint formulas. The price of this simplification would be the additional category check in each of the constraints dealing with the generalized label.

It is often not entirely clear whether a phenomenon is a purely syntactical one (and thus should be dealt with at the syntax level) or involves extrasyntactic considerations. For instance, this model makes no attempt to distinguish between defining and non-defining relative clauses, or to delimit the scope of quantifier expressions explicitly. It is not claimed that the model is the perfect representation of German, or even the best achievable in WCDG. Rather, this overview is just intended to

demonstrate that the parser produces analyses with a considerable amount of detail, and can thus fairly be called a "deep" analyzer.

3.2.4 Subordination strategies

Often the preferable association between words that should be assumed in a dependency analysis is rather clear-cut. For instance, both subjects and objects are tightly coupled with the verbs that they accompany, so that they should form dependency relations, and since one verb can have more than one complement, it should be the regent and its complements, the dependents. Likewise, subclauses are traditionally seen as modifying the proposition of their associated main clause, so that the subclause should be considered the dependent. The grammar model tries to take the obvious choices where possible.

In other cases the association is obvious, but the direction of subordination is less clear. For instance, this model assumes that determiners modify their nouns as dependents, although there are also theories which postulate the inverse relationship, espousing 'determiner phrases' rather than 'noun phrases'. Also, subordinating conjunctions are analysed as modifying the verb of the subclause that they start rather than as the head of the subclause; on the other hand, prepositions are assumed to dominate their embedded nouns rather than modify them like case markers.

Where no linguistic reason can be found to prefer one of several alternatives, a strategy is sometimes chosen because it leads to simpler constraints. For instance, the majority of German verb phrases contain several words because both modal and temporal markers are realized as auxiliary verbs rather than through inflectional variation. Either of the verbs could be chosen as the regent of the verbal complements; however, we choose to subordinate all subjects under the finite verb and all objects and adverbial modifiers under the full verb. This somewhat simplifies the writing of the constraints: agreement rules between the subject and the finite verb can be expressed with a unary constraint, while the type and number of complements should be checked with reference to the full verb, because the full verb determines the possibilities.

Even when no compelling reasons could be found for choosing one of several alternatives over the other, one of them has usually been selected and declared the *only* correct analysis. The reason for this is to avoid *spurious* ambiguity in the model (Komagata, 1997), that is, ambiguity that does not correspond to a clear ambiguity in the propositions that correspond to both versions. For instance, the classic example

"I saw the girl with the telescope."

is commonly used because it allows two structurally different interpretations (they differ in who is holding the telescope). Therefore it is necessary to allow the attachment of the preposition to either the verb or the noun. But the alternatives of

placing the preposition 'with' above or below its associated noun 'telescope' imply no such difference in meaning; therefore, allowing both of them would merely introduce ambiguity into the problem artificially that the parser cannot resolve, thus making analysis unnecessarily hard. The same goes for the placement of determiners.

A more sensitive question especially in the context of dependency syntax is the representation of co-ordinated structures, because not only are there strictly opposing views on how it should be done, but it has even been argued that it cannot be done at all through normal dependency trees (Hesse and Küstner, 1985; Hudson, 1990). A grammar for parsing real-life language cannot afford to adopt this opinion; about 60% of the sentences in the available dependency corpora contain at least one co-ordinated structure. Therefore an asymmetric view of co-ordinations was taken (Mel'cuk, 1987): in asyndetic co-ordinations, the left element is assumed to dominate the right element directly, while each additional element is dominated by the preceding one. When a co-ordinating conjunction is present, it is dominated by the first element with the same label `KON`, while the second element is dominated by the conjunction with the label `CJ`. This decision should not be interpreted as an endorsement of the view that co-ordinations are intrinsically asymmetrical. Both alternatives to the representation of co-ordinations through dependency trees have known weaknesses; one of them had to be chosen in order to avoid spurious ambiguity as described above. There are still complicated 'cluster' co-ordinations which cannot be adequately represented by this model; consider the example:

"Ab dem 1. Oktober beträgt die monatliche Grundgebühr nur noch 24,90 Mark, der Minutenpreis für den Anrufer tagsüber 24, abends 12 Pfennige."
(From October 1, the monthly fee will fall to DM 24.90, the price per minute to DM 0.24 in the daytime and to DM 0.12 in the evening.)
(heiseticker, item 5890)

Three different (though related) predictions are made, but only one verb is used to connect the compared elements; the alternative price options are simply appended without forming a complete sentence, so that this sentence must be analysed as containing several fragments. This is only one instantiation of the more general phenomenon of *ellipsis*. In general, elliptical constructions cannot be treated adequately unless a formalism can postulate empty elements, traces of syntactic transformations or the like. Since WCDG does not support such constructs, elliptical constructions are the greatest source of representational inadequacy in the grammar.

3.3 Implementation

The rules of lexical, structural, and label disambiguation are encoded in the form of about 1,000 binary and unary constraints. Most of these are set down in the file `grammar/deutsch/Grammatik.cdg` in the CDG distribution. The following section will give an overview of the types of conditions implemented, and their method of operation.

3.3.1 Context-sensitive constraints

The precise syntax of constraint declarations will not be reiterated here; see Section 1.4.2 of Foth et al. (2005a), where it is explained in full. However, a general innovation in contrast to previous work in WCDG must be mentioned here. Maruyama (1990) assumed that all well-formedness conditions needed for describing natural language can be encoded into unary and binary constraints, which check one or two dependency relations respectively. An obvious difficulty arose with expressing valence conditions: it cannot be determined whether or not a word is modified by a specific complement unless *all* relations in a syntax tree are checked simultaneously. Maruyama suggested the use of an inverted ancillary level of description just for valence relations, which is coupled to the syntax level with a binary constraint. The presence of the complement on the syntactic level can then be checked with a unary constraint on the ancillary level.

Although this technique suffices for simple valence checks, it quickly becomes cumbersome for realistic grammars. For instance, many words require more than one complement, so that several ancillary levels become necessary. Some constructions even require specific markers that are *not* direct dependents, but merely have to occur somewhere in the subtree of a word (Foth, to appear). Therefore, the WCDG reference implementation was extended to allow operators in constraints which examine the entire *context* of the dependency edge on which the constraint was called. A constraint which uses such an operator is called a *context-sensitive* constraint, while other unary or binary constraints are also called *pure* constraints.

It must be stressed that this new kind of operator is primarily a convenience for the grammar writer and does not per se extend the formal power of WCDG; as Maruyama (1990) showed, his model of CDG was already capable of defining some context-sensitive languages. Although in theory, arbitrary computations could be performed during the call to a context-sensitive operator, they have so far only been used to perform simple checks on the neighbour edges of those named in a constraint signature. Two typical examples that are often used in the grammar discussed here are the new `is()` and `has()` operators.

In its simplest form, a call of the form `is(X^id,L)` checks whether the *regent* of the dependency edge `X` is labelled `L` or not; in other words, it tests the label of the edge *above* the edge `X`. For instance, it is used to enforce the German verb-second condition in main clauses; a simplified version of this constraint is given below:

```
{X/SYN/\Y/SYN} : Vorfeld : order : 0.1 :
 X^cat = VVFIN  ->  ~is(X^id,S);
```

Diagnosing a violation of the verb-second condition would normally require access to three dependency edges: the first pre-modifier of the finite verb, the second pre-modifier, and the subordination of the finite verb itself. However, the third edge is

of interest only with respect to its label: if it reads `S` (main clause), then the co-occurrence of two pre-modifiers should be diagnosed. Rather than extend WCDG to support ternary constraints directly, it is much simpler to write a binary constraint that triggers when two pre-modifiers do in fact occur, and then check the label with `is()`; this check can be done in constant time.

The `has()` operator has the opposite purpose: it checks directly whether or not a word is modified by any edge with a specific label. An obvious application is to express valence conditions succinctly:

```
{X:SYN} : Transitivität : exist : 0.1 :
  X@valence = a  ->  has(X@id, OBJA);
```

This constraint (also very much simplified) demands that any verb marked as taking an accusative object does in fact have one. Obviously this formula is more expensive to compute than the application of `is()` above, since *all* edges must be checked before it can evaluate to 'false' (the answer 'true' might actually be given after the first check). Also, both these and some other context-sensitive operators have more complicated variants; typically they traverse the dependency tree recursively, and check whether a remote regent or dependent satisfies a given condition. However, no context-sensitive operator currently requires more than linear time (in the length of the sentence) to operate.

Although this extension allows the flexible use of nonlocal information without supporting constraints of a higher arity than two directly, it also has disadvantages: it is not always the case that the entire context of a dependency edge is available for checking. This is not a problem for transformation-based local search methods, since they always deal with complete analyses by definition. But a complete search cannot decide whether or not a context-sensitive constraint succeeds, because it typically does *not* operate on entire trees, but on subsets of complete dependency analyses. Therefore the `netsearch` method of WCDG can only evaluate context-sensitive constraints after it has found a complete solution, which could degrade its performance as valuable information about partial results is lost. This trade-off was not made lightly; however, transformation-based local searches have long been the preferred solution methods of WCDG, while the (theoretically) complete search is usually incomplete in practice unless it is allowed to consume excessive time and space. Therefore a further degradation to its effective precision was considered acceptable.

Many of the constraints employed are of considerable complexity. In contrast with the two previous examples, the following examples in this section have *not* been simplified in any way for purposes of exposition; however, those constraints were selected as examples that are self-contained enough to be explained on their own.

3.3.2 Hard constraints

Nearly half of the constraints are *hard* constraints; they describe configurations that should never occur in an analysis. As an example of a very simple hard constraint, here is the definition of what words can be analysed as a PP:

```
{X:SYN} : 'PP-Definition' : init : 0.0 :
  X.label = PP -> isa(X@,Adposition);
```

Note that the 'isa' operator is not a native WCDG construct, but a macro employed in this grammar to abbreviate frequently recurring constructs. The expanded form of this formula is

```
X.label = PP -> subsumes(Features, Adposition, X@cat);
```

which in turn is a more efficient formulation of

```
X.label = PP -> X@cat = APPR |
                X@cat = APPO |
                X@cat = APZR |
                X@cat = APPRART |
                X@cat = PROAV;
```

In other words, this constraint just states that a prepositional phrase must be headed by a word whose category is a type of preposition (the STTS distinguishes prepositions, postpositions, circumpositions and two kind of prepositions fused with words of another category). A *definition constraint* such as this exists for every syntactic edge label, and most of them express similar restrictions on the type of words that can bear that label.

The constraint that describes the possible subordination of prepositions to other words is much more complicated because of the various possibilities. The alternatives are described roughly in their order of perceived frequency:

```
{X:SYN} : 'PP-Unterordnung' : init : 0.0 :
  X.label = PP ->

  // 'Ich wohne<--PP--zu Hause.'
  isa(X^,Verb) |

  // 'Der Alte<--PP--vom Berge sandte seine Assassinen aus.'
  isa(X^,Nominal) |
```

```
// 'Die Echse war starr<--PP--vor Kälte.'
X^cat = ADJA | X^cat = ADJD |

// 'Wo<--PP--zum Teufel steckt der Kerl?'
X^cat = PWAV & X^to = X@from |

// '(Vor allem)--PP-->im Winter ist es sehr feucht.'
isa(X^,Adposition) | X^cat = KOKOM |

// 'Eine<--PP--bis zwei Stunden wird es wohl dauern.'
X@word = bis & (X^cat = ART & X^definite = no | X^cat2 = CARD) |

// 'Mal<--PP--für mal gab es dieselbe Antwort.'
exists(X^cat2) & X^cat2 = NN |

// 'Was<--PP--für ein Pech!'
X\ & X^word = was & X@word = für |

// '(Um 30 Prozent)--PP-->weniger Steuern wurden versprochen.'
X/ & X@word = um & exists(X^degree) & X^degree = comparative;
```

In this constraint, the expression `X@from` denotes the time point at which the dependent of the edge `X` starts, while `X^to` is the time point at which it regent ends (in written input, the first word is defined to start at time point 0 and end at time point 1, and so forth). The formula `X^to = X@from` therefore expresses that `X` relates to directly adjacent words.

The most prevalent subordination is that of prepositions to verbs or nouns. Adjectives can also be modified by prepositions. All other cases are comparatively rare phenomena. For instance, the expression 'wo zum Teufel' ideosyncratically allows an interrogative pronoun to be modified by the preposition 'zu', therefore this combination has to be allowed (note that the English equivalent 'what the hell' contains a similar abnormal category combination). Also, some fixed phrases that are syntactically prepositional phrases are used with a general adverbial sense and can modify even other adverbials; thus, the expression 'vor allem im Winter' merely means 'especially in winter', although the word 'vor' usually has a temporal or local sense. Several other exceptional constructions are also explicitly licensed such as 'eine bis zwei Stunden' ('one or two hours') or 'um 30 Prozent weniger' ('30% less').

It can clearly be seen how this type of constraint becomes progressively longer as the grammar extends to deal with more cases and even idiomatic constructions. This is necessary to make sure that all cases considered correct are covered, and only those. Even when exceptions pile up so that ultimately most categories can be modified by `PP` dependencies under some circumstances, it would not be appropriate to simply

omit the subordination constraint; instead the possible cases should be described and restricted where this is possible. For instance, the construction [preposition] + [quantified expression] + [comparative expression] is only possible with the preposition 'um', and therefore should not be generally allowed. That this is enforced by a hard constraint does not mean that an expression such as **bei 30 Prozent weniger*, which violates the lexical condition, cannot be analysed — the preposition could still be assigned a different label, or subordinated somewhere else. As a last resort, it could always form an uninterpretable fragment.

A drawback of the semantics of the constraint language as used by WCDG is the fact that there is no better way do deal with this kind of exception. The multiplicative combination policy dictates that what is disallowed by a hard constraint can never be allowed again by any other constraint; therefore, an exception to a hard rule like this *must* be formulated within the rule itself and not in a separate constraint. This often leads to definition constraints that are so littered with exception specifications that the fundamental regularity becomes quite obscured.

A possible alternative would be to leave out all of the exceptional cases and instead make the constraint defeasible instead of hard. This would prefer the normal behaviour of prepositions where this is possible, but still allow the analysis of exceptional cases. However, this would allow *all* regents uniformly, even those that cannot possibly be correct (such as articles or even punctuation tokens). It would also treat the idiomatic expressions as merely tolerated aberrations, where a native speaker would consider them perfectly correct, even preferable, despite their unusual category combinations. Therefore the general strategy in this grammar was always to describe the possible cases as closely as possible, so that the truly unwanted cases can be excluded as early as possible by hard constraints.

3.3.3 Defeasible constraints

Constraints that describe defeasible knowledge can vary from nearly universal principles to vague preferences. An example of a nearly universal constraint is the following *normalization constraint*:

```
{X\SYN/\Y\SYN} : 'Adverb am Hilfsverb' : category : 0.1 :
  X.label = AUX & isa(X@, Verb)
  ->
  ~edge(Y,Adverbiale_Unterordnung);
```

The operator '`edge`' is a macro defined by the grammar, just like the '`isa`' described above, but checks edge labels rather than word categories. The expansion of the conclusion of this constraint would be

```
  ~(Y.label = ADV | Y.label = PP | Y.label = ZEIT | Y.label = KOM)
```

Normalization constraints are used to contain spurious ambiguity. As discussed in Section 3.2.4, adverbial modifiers in the middle field should normally modify the full verb, not because this is intrinsically better than the alternative, but because we have arbitrarily decided upon this policy. This rule is formulated as defeasible because there are some very rare circumstances in which this policy cannot be followed consistently, and which would be too difficult to describe as an exception. For instance, in the following example an entire subclause intervenes between the prepositional phrase and the full verb, so that it is forced to attach to the final verb instead:

"Die Bundesregierung kann im Verteidigungsfalle, soweit es die Verhältnisse erfordern, 1. den Bundesgrenzschutz im gesamten Bundesgebiete einsetzen."
(During a state of war, the Federal Government may, if conditions necessitate this, 1. employ the Federal Border Guard in the entire area of the Federation.)
(Grundgesetz §115f)

It is somewhat unusual for a normalization constraint to be defeasible; most normalization rules can be written as hard constraints. The defeasible rather but strict constraints usually deal with grammatical rules that are considered to hold, but whose violation does not necessarily prohibit understanding. For instance, German subjects and predicative complements are not distinguished by case; therefore the general tendency is to mark the subject by placing it to the left of the predicative. This rule encodes the regularity for the case that both appear in the middle field, where it holds almost universally:

```
{X/SYN/\Y/SYN} : 'PRED-SUBJ-Reihenfolge' : order : 0.1 :
  X.label = PRED & isa(X@,Nominal) &
  Y.label = SUBJ
  ->
  distance(X@id, Y@) < 1 |
  isa(X@,Pronomen) |
  initial(X@);
```

Note that even this regularity conflicts with two other, more important ones: pronominal noun phrases should generally precede non-pronominal ones, no matter what their grammatical function, and noun phrases which contain interrogative or relative pronouns must form the first element of a subclause no matter what. In these circumstances, other, more important constraints regularly overrule this one; this alone would be enough to define the correct analysis of relative clauses, but this constraint also makes the exceptions explicitly, in order to avoid unnecessary contention between constraints.

Preference constraints encode generalizations that often hold but are easily overridden by other factors.

```
{X:SYN/\Y:SYN} : 'Subjekt-Position' : order : 0.9 :
```

```
X.label = SUBJ & edge(Y,Nominalobjekt)
->
X@from < Y@from |
Y.label = OBJD |
Y@cat = PRF |
initial(Y@);
```

This constraint restricts the order of co-occurring subjects and other nominal objects. German prefers the order SVO, but in contrast to English this preference is very weak (see Section 2.7.3), and therefore the penalty for violating it is much closer to 1. For instance, in the following sentence

"Gleiches melden uns Leser aus dem ganzen Bundesgebiet."
(Readers from all over the country make similar reports.)
(heiseticker, item 3770)

the more important agreement conditions between subject and verb easily override the normal ordering; native speakers interpret this sentence without even being aware of the inversion. Nevertheless, the rule, while defeasible, is not at all unimportant. Where no other syntactic clues are available, speakers of German assume the default ordering quite reliably:

"Die Verfassung lässt die Eigentumsordnung in den verschiedenen Mitgliedstaaten unberührt."
(The Constitution leaves unaffected the property regulations in the various member states.)
(European Constitution, §III-425)

The question of whether the European Constitution or the preexisting local laws ultimately take precedence in property disputes is eminently important, nevertheless the disambiguation is entirely entrusted to the subject/object ordering preference.

Finally, there are constraints whose penalty depends on features of the edge that violates them. Typically, an edge is dispreferred the more the longer it is:

```
{X!REF} : 'REF-Distanz' : dist : gradient(40) :
  abs(distance(X@id, X^id)) < 3;
```

Here, the operator '`gradient`' is another macro defined by the grammar that specifies a variable penalty that depends on the length of the dependency edge in question. The expression `gradient(40)` expands ultimately to the formula

$$40/(40 + \text{abs}(\text{length}(\text{X}))$$

The constraint expresses a rather stringent condition that the distance between a relative pronoun and its antecedent should be small. For instance, the following

example is difficult to understand because 10 words intervene between the words 'T-DSL-Zugangs' and 'der':

"Die Einführung des T-DSL-Zugangs für 9,90 Mark als Zusatzoption zu ISDN-300- oder ISDN-XXL-Anschlüssen, der sich mit T-Online flat kombinieren lassen soll, scheint für den 1. August geplant zu sein."
(Introduction of the T-DSL rate for DM 9.90 on top of ISDN-300 or ISDN-XXL lines, which can be combined with T-online flat, is apparently planned for August 1.)
(heiseticker, item 9833)

Most other distance constraints are weaker than this one, since a great linear distance between regent and dependent is much more common for other relation types. For instance, the `AUX` relation typically connects words that are quite distant because they delimit the entire *Mittelfeld* consequently, the corresponding constraint has a gradient of 1000 rather than 40.

3.4 Operation

3.4.1 The numerical model

Recall that the penalty of a constraint contributes to the score of an analysis only if the analysis contradicts the constraint — more than once if more than one part of the analysis contradicts it. The score of an analysis is the product of all these individual penalties (or 1 if all constraints are satisfied). Of all possible analyses of a given sentence, the parser aims to construct the one with the highest score. As an example, we shall now consider in detail the analysis of the following sentence

"Niemand darf gegen sein Gewissen zum Kriegsdienst mit der Waffe gezwungen werden."
(No one must be forced into armed service against his conscience.)
(Grundgesetz, §4)

See Figure 3.1 for the corresponding dependency tree that corresponds to the intended meaning. This analysis is not only the numerically optimal structure as defined by the constraints, but it also corresponds to the reading of the sentence preferred by an informed human reader. We will now give an informal argument why no other analysis can exist that scores higher; the names of the constraints which enforce the conditions cited will be given in parentheses.

To begin with, the finite auxiliary 'darf' requires an infinitive as a complement ('Transitivität'). There are numerous exceptions to this general rule, but none of them applies to this analysis. For instance, a verb may lose its complement if it is co-ordinated and the complement modifies the left conjunct instead; but no conjunction occurs in this sentence, and there is no preceding verb which could serve to form an asyndetic co-ordination. The valence frame of the verb could also be changed by a

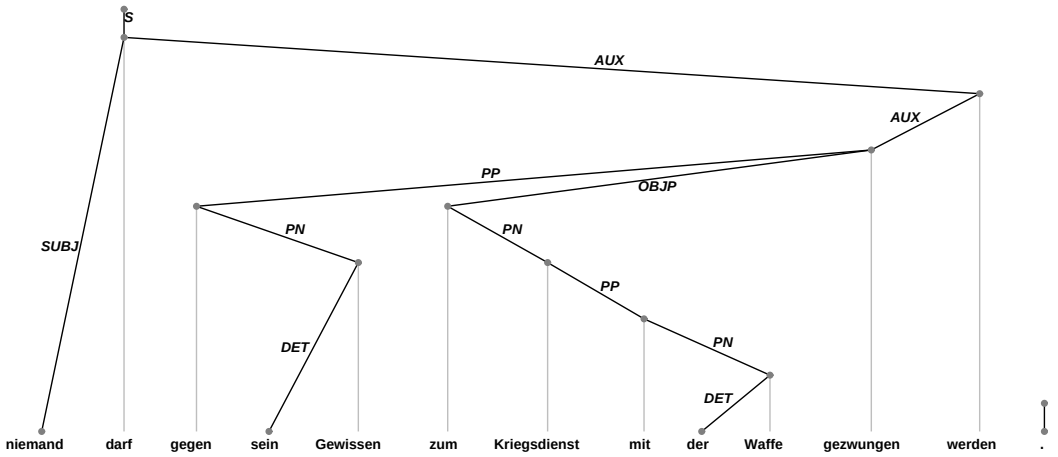

Figure 3.1: Analysis of a sentence from German law text.

separable prefix: for instance, the modal verb ‘darf’ takes an auxiliary complement, but loses this valency when modified by the particle ‘heran’. But again, the utterance contains no word with the fitting category `PTKVZ` that could effect such a change. Likewise, the valence for a direct object disappears in a passive construction, but ‘darf’ is manifestly in the active voice.

The only infinitive in the sentence is the form ‘werden’. It follows that any analysis *must* either subordinate it under ‘darf’ with the label `AUX`, or incur the transitivity penalty. Much the same reasoning holds for the verb ‘werden’ itself; the only way of satisfying its requirement for a participle complement is to attach the preceding ‘gezwungen’ to ‘werden’, also with the label `AUX`.

Next, consider the three prepositions in the sentence. Each of them requires a nominal complement (‘PN fehlt’). Again, the requirement could be avoided if several of them formed a co-ordination, but asyndetic co-ordination between prepositions is not allowed without some sort of marker such as a comma (‘Komma für KON fehlt’). Each of the prepositions must therefore dominate one of the three nouns, or cause another large penalty. Definition constraints (‘PN-Stellung’) require that a preposition precedes its complement, therefore the pairing of nouns and prepositions can only be as illustrated.

Both of the determiners in the sentence are now within the scope of a preposition-complement relation. General rules of tree structure (‘Projektivität’) prevent a determiner from modifying any word outside this scope, and category constraints (‘DET-Definition’) forbid attachment to a preposition. Determiners might also refuse to subordinate altogether and form a fragment instead, but a separate constraint (‘Fragment’) punishes this possibility even more severely than any requirement constraint. Therefore both determiners must modify the following word to avoid all these penalties.

Since all three nouns modify prepositions, none of them is available as a subject for the finite verb. The only remaining possibility without incurring another severe penalty ('Subjekt fehlt') is to make the pronoun 'niemand' the subject; this is also the only reasonable attachment left for that word, which would otherwise cause another fragment penalty.

The last remaining ambiguity is where to subordinate the three prepositions in the sentence. In general this is a notoriously difficult problem whose solution requires information from many different levels of linguistic description. In this case, however, the existing constraints suffice to make the correct decision. Each of the prepositions might either modify the verb phrase or a preceding noun. A default rule ('Präpositionalattribut') specifies that, *ceteris paribus*, a subordination under a verb is preferred to one under a noun. Normalization constraints ('Adverb am Hilfsverb') ensure that if the attachment is to the verb phrase, the full verb rather than the auxiliaries should be modified, so that two of three options for verb attachment can be discounted. All three prepositions, then, could modify the word 'gezwungen' and so complete a spanning tree that violates no important constraints.

The first preposition in fact has no other possibility than to modify the verb phrase as a 'PP'. The preposition 'zu' could either do the same or modify the preceding noun; the default rule mentioned above would alone suffice to prefer the verb attachment. However, there is another and more important relevant rule: the lexicon specifies that the preposition 'zu' (unlike both other prepositions) can form a prepositional object for the verb 'zwingen'. This is preferable both because it satisfies the (optional) valence of the verb, and because adjuncts receive a further small penalty ('PP could be OBJP') if they are homonymous to object prepositionals, since for possibly idiomatic constructions, the idiomatic meaning is more likely than the non-idiomatic one.

A final consideration is now that if the last preposition 'mit' also modified the verb, it would follow its sibling object preposition; this is dispreferred ('OBJP vor PP') because complements generally appear closer to their regents than adjuncts. This dispreference is greater than the default rule against prepositions modifying nouns, and so the very best score is achieved by attaching 'mit' to the preceding noun.

We mention in passing some complications that arise in the actual implementation of this optimization problem. First and foremost, this discussion has tacitly assumed that all of the words do in fact belong to the category that the preferred analysis assigns to them. If, for instance, the word 'werden' were analysed as its homonymous finite variant, the entire chain of reasoning above would fall. We will see in Section 3.6 that this is in fact a major problem in practice. Section 4.1 describes how to resolve category ambiguity; the methods introduced there are sufficient to let this analysis (and most others) come out the preferred way.

There are also morphological variants below the category level for several words. However, all of these are resolved by agreement constraints: for instance, the preposition 'gegen' forces the determiner 'sein' of its dependent into the accusative case

rather than the homonymous nominative ('DET-NP-Kasus'). In a similar way, the neuter noun 'Gewissen' selects the neuter variant of the word 'sein', which might otherwise be masculine ('DET-Genus'). This word, in turn, is unambiguously a singular determiner, and therefore establishes that the word 'Gewissen' is a singular form as well, although it has a plural homonym ('DET-Numerus').

Finally, we have glossed over the possibility of extrasyntactic reference dependencies; these cannot not occur ('REF-Definition') because there are no relative pronouns in the sentence. Despite these minor complications, the grammar (in combination with part-of-speech tagging) successfully defines a unique structure that scores at least slightly better than any alternative, and corresponds accurately to the preferred reading of the sentence.

This discussion is not intended to give the impression that the described model of German is always as accurate as in this example. On the contrary, it contains various systematic weaknesses; see Section 3.6 for a fuller discussion. For instance, note that the subordination of the final preposition discussed above creates an ambiguity of meaning as well as structure: attaching it to the word 'gezwungen' after all would change the meaning of the sentence to "No one may be forced at gunpoint to join the army against their conscience." By itself this is a plausible reading, since many people have indeed been forced at gunpoint to join an army; but the authors of this constitution doubtlessly meant that no one may be forced, by *any* means, into armed service, since that might violate their conscience. Therefore the purely syntactical distinction between object and adjunct prepositions does the right thing in *this* example; but it does so not through duplicating this last argument of the chain, since the necessary reasoning about background knowledge is far beyond the capability of the model. It is merely because this sentence follows the syntactic preference rather than violate it that the grammar rules correctly predict even this last attachment; it is easy to construct examples where the same rule causes a wrong subordination to be selected.

It should also be stressed that the type of demonstration of optimality sketched here does *not* directly correspond to any algorithm in the WCDG reference implementation. The fundamental process of weighing the violation of different constraints against each other is the same; but in our discussion, we cleverly considered only few words at a time, concluding that the optimal analysis would have to include a particular dependency. The conclusion then allowed us to dismiss many alternatives for the next subordination, and so on. Choosing the correct order in which to exclude the possible attachments was essential to arriving at a solution quickly. But this requires a process of meta-reasoning that is far more difficult than writing the grammar itself, even though it seems easy to those who are fluent speakers of the target language. Instead, all currently implemented algorithms conduct variants of a complete or incomplete search whose heuristic components are far simpler. Although this often has the same result (as in fact it does when the example sentence

is actually parsed), it can lead to search errors even in analyses where common sense suggests that the analysis should be easy.

3.4.2 The computational model

Figure 3.2 shows several intermediate results of the transformation algorithm when it is actually applied to the example sentence. The first analysis is the starting point of the transformation process; it was created by simply choosing the value with the highest unary score from each domain. As is to be expected, the resulting dependency tree violates many rules that cannot be formulated as unary constraints: it is not even a proper tree, since the last two words are subordinated under each other; the words 'darf' and 'gezwungen' both have multiple objects with the same label, which is not allowed; it postulates a direct object for a verb that is marked as a passive form, which never happens, and so forth. Usually, the first few transformation steps are dedicated to removing such obvious nonsensical combinations.

The second tree shows the first intermediate result that carries a score larger than 0; in other words, this is the first analysis that might (in theory) actually be correct. Certainly it looks much more like a sensible dependency tree in that it assigns a plausible interpretation to the first half of the sentence. However, despite great improvement in terms of the numerical model, there has been no progress at all in terms of accuracy: this analysis predicts the correct regent for 7 of the 13 tokens, but makes errors for 6 tokens; this is exactly the the same figure as for the nonsensical starting point, while the *labelled* accuracy has actually fallen by 1.

The worst conflict in the second tree is the lack of a complement for the preposition 'mit', and therefore this conflict is scheduled for correction next. The next step shows how the neighbouring noun is recruited to fill the valency; since this is in fact the correct subordination, the number of structural errors drops to 5. Now the worst conflict is the treatment of the full verb as a fragment.

The fourth tree shows how the parser manages to remove this conflict by interpreting the passive participle as an adverbial modifier. This construction is somewhat dispreferred and therefore introduces a new conflict, but is far preferable to a fragmented syntax structure, and so a new numerical optimum is achieved although the new subordination is no more correct than the old one. From our discussion above, we know that it would have been better to change the subordination to 'werden' rather than to 'Waffe', but repairing the entire verb phrase would require a longer sequence of steps before a numerical improvement is found. The algorithm took the smaller improvement because it could be reached directly, while the correct decision would have lead to a longer sequence of steps against the hill climbing criterion; the method never does this unless forced to by the tabu criterion.

Although the adverbial interpretation that was just introduced leads to a globally better score, it does so at the cost of introducing a new conflict. In fact, this conflict

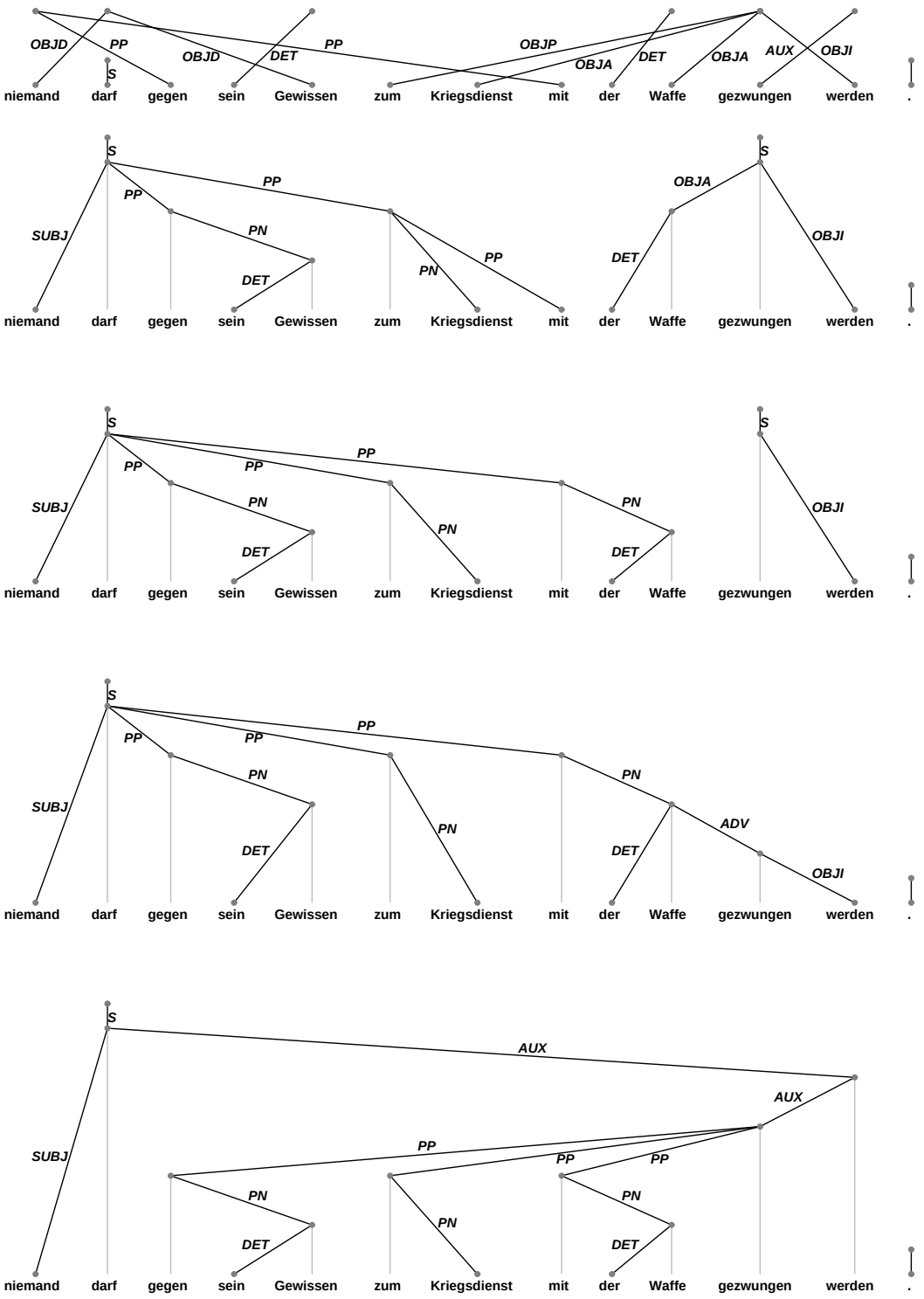

Figure 3.2: Intermediate results during transformational analysis of the same sentence.

causes the worst penalty of all remaining ones. Therefore the algorithm immediately tries to change this construction again. This time it *is* forced to take the longer sequence of steps that was rejected earlier, and find a new subordination for 'gezwungen'. Note that five changes are necessary until a numerically better analysis (the fifth tree) is found; this involves lowering all three prepositions to satisfy the verb phrase normalization constraint mentioned earlier.

The analysis is now correct except for the word 'mit'. At this point, the parser entertains precisely the alternative interpretation 'forced at gunpoint' considered in Section 3.4.1. The ordering preference between modifiers and adjuncts mentioned there is now the worst conflict in the analysis. The parser duly looks for a way to avoid it, and soon finds the correct analysis as in Figure 3.1; and so even the last error is corrected.

3.5 Limitations

Independent of processing difficulties such as lack of resources, which could at least theoretically be overcome by allowing more processing time and space, there are still some areas of language that are not satisfactorily addressed even in the declarative model. Perhaps the most relevant of these for practical application would be the question of *dialectal* and individual variation. Phenomena such as reordering or lexical selection are dealt with by graded constraints and lexicon templates[4]; but dialectal variation can gradually transform an entire language, by mutually comprehensible shifts, into a totally different one. At some point a fixed grammar must abandon the goal of covering all possible variations, and be prepared to declare utterances to be not sufficiently standard language to be analysed.

For instance, the phrase traditionally spoken at carnival sessions, "Wolle mer neilasse?" would, in the standard dialect of German, be spelt "Wollen wir ihn hereinlassen?" *(Shall we let him in?)*. This phrase cannot be adequately analysed; neither of the closed-class words is in a form that the lexicon recognizes, and hypothesizing new closed-class lexicon items is not attempted. The infinitive has been elided to the point that it does no longer bear the characteristic infinitive ending '-en', and therefore it can likewise not be analysed even as an unknown verb. Although a limited kind of support for such speech could be achieved by extending the lexicon, the task would be open-ended. At the moment it can be analysed merely as a string of foreign-language material.

A different kind of borderline case are transcripts such as the ones of plenary sessions of the German national parliament. Although the representatives' speeches contain well-designed language, the corpus overall is full of interjections, abbreviations and

[4]For instance, see Section 3.6.3 for performance on biblical German, which obeys systematically different ordering rules.

isolated noun phrases which are of little interest, and overlap in a way that makes it difficult even to delimit the individual sentences uttered:

> Joachim Hörster (CDU/CSU): Frau Präsidentin! Meine Damen und Herren! Wir haben eben bei der Beantwortung der dringlichen Fragen durch die Bundesregierung nicht ausreichende Antworten erhalten, bezogen auf das Thema, um das es geht. (Lachen bei der SPD sowie bei Abgeordneten des BÜNDNISSES 90/DIE GRÜNEN und der PDS) Wir wissen seit Monaten, (Freimut Duve [SPD]: Genau!) daß es einen Vermerk des Ehrenvorsitzenden der Sozialdemokratischen Partei gibt, (Freimut Duve [SPD]: Und dann der Regierung unangemessen!) der in den Geschäftsgang der Parteiorganisation der SPD gegangen ist, (Freimut Duve [SPD]: Ohne daß die Regierung das weiß!) der Hans-Jochen Vogel veranlaßt haben soll, (Joseph Fischer [Frankfurt] [BÜNDNIS 90/DIE GRÜNEN]: Unglaublich!) mit dem Präsidenten des Bundesnachrichtendienstes ein Gespräch zu führen.

A similar effect occurs when tabular material is for some reason included into running text:

> Intel Coppermine 600 MHz (P III-600EB, 4,5 x 133) $455 Intel Coppermine 600 MHz (P III-600E, 6 x 100) $455 Intel Pentium-III 600 MHz (P III-600B, 4,5 x 133) $465 Intel Pentium-III 600 MHz (P III-600, 6 x 100) $465 AMD Athlon 600 MHz (6 x 100) $419

Such language can be analysed as a long string of noun phrases, but in the end there is little point in analysing it at all; the data are presented much better in their original tabular form.

3.6 Evaluation of the current state

A purely linguistic model of a language can be evaluated on qualities such as perspicuity, coverage, or predictive power. However, a computational model must also be able to prove its ability to analyse new text successfully in order to be considered practically useful. This section has a threefold purpose: existing work in broad-coverage parsing of German is reviewed; our own system is applied to similar tasks, and its accuracy is compared to previously published results; and those errors that our parser makes are investigated more closely, particularly with respect to possible improvements.

3.6.1 Related work

Most systematic evaluations of automatic parser have focused on the processing of English, and even a particular corpus of English: it has become almost mandatory to run a system on the Wall Street Journal section of the Penn Treebank (Bies et al., 1995) and report the results. For statistical models, even the partitioning of this corpus into training, development and testing data has become ritualized: approaches as diverse as Magerman (1995); Eisner (1996); Collins (1999); Charniak (2000); Clark et al. (2001); Klein and Manning (2003); Henderson (2003) all use the same setup. This has the obvious advantage that the proposed algorithms can be directly compared with respect to their success. On the other hand, it means that the majority of world-wide parsing research has been conducted on only a single language: the variety of English used in the Wall Street Journal to report on business transactions. Also, the particular formalization as nested constituency structures has repeatedly been criticized as limiting and not rich enough for useful applications of parsing. As a result, many parsers that solve the standard task well are not necessarily very good at parsing other texts (Gildea, 2001).

There is no standard task for German parser evaluation that is as rigidly defined as this; in fact, there are very few evaluations of broad-coverage parsers of German at all. In an early attempt to construct a general parser of German from a tree bank, Fissaha et al. (2003) extracted an unlexicalized PCFG from the first 18,000 sentences of the NEGRA corpus. Through the judicious use of parent encoding (the projection of the category and function of a phrasal node onto the pre-terminal nodes), f-scores around 70% of all constituents were achieved on the test set, however, coverage was only 70% of all sentences. Dubey and Keller (2003) reimplemented the statistical model from Collins (1999) and parsed the next-to-last 1,000 sentences of the NEGRA corpus with it[5]. They found that this model is much less successful on the German corpus than it was for the Penn Treebank: the labelled recall and precision of NEGRA constituents was only 68.6%/66.9% rather than the figures near 90% so often published for English, even though the NEGRA sentences are 25% shorter on average.

There could be many reasons for these disappointing results; both the properties of the language itself and of the particular annotation model espoused by the NEGRA specification could be at fault. For instance, PCFG parsers are systematically more suited to configurational languages than to languages such as German in which considerably more freedom is possible in word order (Levy and Manning, 2004); a WSJ parser can assume that any preverbal noun phrase is a subject without making many errors, while in German this assumption is quite inappropriate.

On the other hand, the details of the phrase structure as specified in the NEGRA annotation specification could be subtly different, so that the same similarity model does not work equally well on them. A model that was developed by repeatedly

[5]Only the 968 sentences shorter than 41 words were analysed.

testing it on the WSJ corpus, even on a held-out test set, might have developed so as to take maximum advantage of the information in that type of tree structure. This effect is perhaps not very surprising, but it should be kept in mind, since one of the major points usually advanced in favor of statistical parsers is that they can be easily retrained for new languages, given enough data; these results show how far real-life results can fall behind this ideal, even when porting between languages such as German and English, which are quite closely related as languages go.

There is probably some truth to both explanations. However, a broad-coverage parsing system cannot change the language itself to suit its own limitations, and therefore only the second one can be leveraged to improve the results. Dubey and Keller (2003) in fact concluded that the particular conditional model developed for the Penn Treebank is systematically suboptimal for analysing NEGRA phrase structure; the rules in the NEGRA corpus are said to be substantially 'flatter', so that they are less suited to be captured by the lexical head-to-head bigrams that Collins defined. They replaced these bigrams with sister-to-head bigrams instead, and raised the result to 73.9%/74.2%. In later work (Dubey, 2005) a labelled f-measure of 76.2% was achieved through sophisticated smoothing methods.

Schiehlen (2004) parsed the same set of sentences and solved the problem in a different way: rather than adapting the similarity model of his parser, he systematically transformed the tree bank in ways that allow the model to learn salient differences better. The parser was then trained on the enhanced versions of the training set and applied to the test set; its output was subjected to the inverse transformation, and finally compared to the original trees in the test set. Generally, these transformations introduced finer subdivisions of the set of phrase node labels. For instance, German can express possession both with the genitive case or with the preposition 'von'; when those prepositional phrases that actually express possession rather than locality were given a special category label, accuracy improved. Altogether, the parser improved from an f-score of 78.1% to 81.7% when measuring the task of building dependency structure labelled by the grammatical functions of NEGRA.

Most other published results cannot be directly compared to our experiments because they either do not address the full parsing task, or do not report the accuracy of their results. For instance, Langer (2001) states only that the 'Gepard' system can analyse 33.5% of all sentences from authentic newspaper articles, but not how faithful the analysis is. Braun (2003) reports a precision/recall of 86.7% and 87.3% respectively on a test set of 400 sentences, but only for the simpler task of determining the boundaries of topological fields.

3.6.2 Experiment

We shall now review the performance of our WCDG of German as it is. Although the NEGRA corpus is no longer the largest corpus of German available (Brants et al., 2002), for purposes of comparison we follow the examples cited in Section 3.6.1 and

take the dependency versions (see Section 3.1) of sentences 18602 through 19601 of the NEGRA corpus as our test set. Each sentence in the test set is analysed by transformation-based search as described in Section 2.5.3. To minimize the amount of search errors in the results, we set an extremely long timeout value of 1,000 seconds for each parsing run; this is enough for over 90% of all runs to terminate on their own, usually long before the time limit.

In the interests of comparability, for the time being we also retain the assumption of Fissaha et al. (2003) and Dubey and Keller (2003) that the syntactic category of each token is known in advance. This is of course unrealistic, particularly for a grammar intended to provide very broad coverage; methods to replace this unwarranted assumption with a stochastic predictor will be developed in Section 4.1.4.

		Accuracy	
Length	Instances	structural	labelled
1 – 10	340	95.0%	93.4%
11 – 20	323	91.7%	90.0%
21 – 30	229	90.7%	89.3%
31 – 40	76	87.3%	85.8%
≤40	968	90.9%	89.4%
>40	32	84.7%	83.4%
overall	1,000	90.4%	88.8%

Table 3.2: Results of applying the handwritten grammar to NEGRA test set.

The results of this initial experiment are given in Table 3.2. Out of the 16,690 tokens in the test set, 15,068 are assigned the correct regent; 14,829 of these also receive the correct edge label. This corresponds to a structural accuracy of 90.4% and a labelled accuracy of 88.8%.

3.6.3 Comparison

An exact comparison with previous work is difficult for several reasons. First of all, the parsers described in Fissaha et al. (2003); Dubey and Keller (2003) operate on phrase structures, while we run and evaluate on dependency edges. As discussed in Section 2.6, a comparison of accuracy figures between dependency and constituent recall or precision is not necessarily meaningful even for the same corpus, and even though dependency and constituent structure are in principle similar in their expressiveness. The task solved by either parser might be harder or easier, depending on details of the annotation guidelines.

It should be noted that we perform dependency evaluation because this is the natural thing to do for a parser that operates in dependency space, and not in order to avoid the mismatch of the PARSEVAL metrics to parsers with partial (Kübler and Telljohann, 2002) or particularly rich structures (Bangalore et al., 1996). Since the

NEGRA corpus was intentionally written to contain rather flat phrases (Skut et al., 1997), it does not encode many of the subtle structural distinctions that e.g. a full X-bar representation could make. This could indicate that retrieving its structures is not as different from directly constructing word-to-word subordinations as it could be; however, this could only be confirmed by control experiments in which the *same* text is analysed with a PCFG parser under different annotation schemes. Finally, it should be noted that we parse all 1,000 sentences, and not just the ones shorter than 41 words, so the most closely comparable figures from our experiments would be the labelled accuracy of 89.4% and not the overall 88.8%.

In contrast, the work by Schiehlen (2004) analyses all 1,000 sentences; it also follows Lin (1995) in using word-to-word dependencies for evaluation, although the parser operates in phrase structure space. However, a unique analysis is not guaranteed for all input, so that precision and recall are reported rather than only the accuracy; therefore we can only compare our accuracy of 88.8% to the reported dependency f-score of 81.7%. On the other hand, his experiment did not assume that syntactic categories are known in advance. We will show in Section 4.1 that this distinction explains only a small part of the difference.

A second problem is that the experiment we conducted does not accurately measure the accuracy of the WCDG model itself; as discussed in Section 2.7.3, search errors resulting from the incomplete solution methods employed will distort the measurements. In contrast, most statistical language models do allow a complete search to be performed, and indeed are chosen with this property in mind.

Since there are many more wrong than correct assignments, it is to be expected that search errors will decrease rather than increase the measured accuracy. Therefore the model itself can be expected to be somewhat more accurate than our measurements indicate. However, they correctly describe the performance of the complete WCDG system, including grammar and search methods, and are therefore adequate for comparing its practical use to other parsers. We will reconsider the distinction in Section 3.6.4 when discussing possible improvements to the entire system.

Perhaps the most important barrier to a direct comparison is the fact that WCDG differs from all these systems in its different methodology: in the terms of Bangalore et al. (1995), it is a *hand-crafted wide-coverage grammar*, while the other approaches were *statistically induced corpus grammars*. It is not entirely clear whether such different systems can be meaningfully compared by a direct juxtaposition of accuracy figures. On the one hand, a grammar learnt from a particular corpus is by definition corpus-dependent; it can be expected to achieve its best results on the same corpus, provided that the corpus is homogeneous enough that regularities from the training set recur in the test set. Because a wide-coverage grammar is expected to deal with different domains and text types, its task is more difficult, and it should be expected to be at a disadvantage when compared against a corpus-dependent parser.

On the other hand, a hand-written parser might enjoy a different subtle advantage. A statistically induced parser is expected to observe a strict discipline in distinguishing

between training, development, and testing material (Callison-Burch and Osborne, 2003). The system should be written so that it can learn properties observed in the *training set*, and variants of the similarity model should be selected by optimizing performance on the *development set*, while the published results should be measured on a *test set* that is disjunct from either. This ensures that the parser makes useful generalizations about its observations instead of simply memorizing what it sees. Of course, research does not stop after measuring: for instance, the decision of Dubey and Keller (2003) to switch the type of the bigrams used for conditioning might be regarded as infringing upon this separation, since it was inspired by results on the test set, and not the development set, but if so, it was clearly a minor and necessary breach.

A hand-written parser cannot observe the same strict discipline, since corpus acquisition, annotation specification, and rule development all take place in the same brain. In particular, the NEGRA corpus was already known to the author when the rules of the WCDG of German were developed. In an expressive formalism such as WCDG it would have been very easy to tune the grammar rules so that they perform especially well on the intended test set. In theory, rules could even be written that detect specific features of individual sentences, and enforce the correct analysis for them; but even without such blatant cheating, unconscious bias could very well have similar effects. In other words, it must be verified that the advantages of greater expressivity that a handwritten system brings were not used to compromise its general applicability.

To prove that the WCDG of German is not simply a NEGRA test set grammar in disguise, it would in principle be sufficient to declare that the constraints themselves are available for public inspection, and that none of them are tailored particularly towards constructions from this corpus. We do in fact claim precisely that. Nevertheless, some further points should be made:

- Regular development work did not use the NEGRA corpus, but the other sections of the dependency corpus, mainly the 'heiseticker' sentences.
- The DEPSY tool used for converting NEGRA structures into dependency trees was developed on the first 3,000 sentences of NEGRA, and then applied to the test set. Although its output was manually reviewed to ensure they conform to the WCDG model as well as possible, fewer than 1% of all subordinations were changed during this post-editing, so that this stage cannot introduce major bias.
- No part of the specification of German was changed to facilitate NEGRA parsing. For instance, labels in the NEGRA corpus without a direct counterpart in our model, such as the `RE` label for 'repeated elements', were not added to the label set, but mapped to structures already covered by the grammar. The WCDG label 'OBJP' to distinguish object from adjunct prepositions was introduced before its addition to the TIGER annotation guidelines.

The most decisive evidence that the system presented here is a *broad-coverage* parser is of course obtained by applying it, unchanged, to other corpora. In order to provide this evidence, four other sections of our dependency corpus were parsed under precisely the same conditions as in the previous experiment:

1. The 2003 revision of the German federal constitution (*Grundgesetz*)
2. The first 10,000 sentences of online newscasts (*heiseticker*)
3. The book of Genesis as translated into German in 1960 (*Genesis*)
4. 9,547 sentences from a contemporary fantasy novel (*wyvern*)

			Accuracy	
Corpus	Length	Instances	structural	labelled
Grundgesetz:	1 – 10	409	97.2%	95.6%
	11 – 20	327	96.0%	95.0%
	21 – 30	221	92.7%	92.0%
	31 – 40	95	88.6%	87.6%
	>40	102	83.1%	81.6%
overall	18.4	1,154	90.7%	89.6%
heiseticker:	1 – 10	2,519	95.0%	94.0%
	11 – 20	3,976	93.8%	93.0%
	21 – 30	2,499	91.7%	90.6%
	31 – 40	806	89.3%	87.7%
	>40	200	85.4%	83.6%
overall	17.3	10,000	92.0%	90.9%
Genesis:	1 – 10	1,106	97.1%	95.7%
	11 – 20	873	94.2%	92.4%
	21 – 30	456	92.4%	90.5%
	31 – 40	169	91.1%	89.1%
	>40	105	88.4%	85.6%
overall	15.9	2,709	93.0%	91.2%
wyvern:	1 – 10	3,905	97.2%	95.2%
	11 – 20	3,966	94.8%	92.9%
	21 – 30	1,299	92.7%	91.0%
	31 – 40	295	89.7%	87.5%
	>40	82	86.7%	84.1%
overall	13.8	9,547	94.2%	92.3%

Table 3.3: Parsing results on other text types.

Table 3.3 shows the results of these control experiments. The foremost observation to be made is that other text types can be parsed with similar success; the attachment

accuracy is above 90% in all cases. There is also a clear trend that longer sentences are more difficult to parse, both within each corpus and when comparing between them. However, parsing difficulty could also be related to different predominant constructions in different types as well as to sheer length; it seems plausible that law texts intended to be understood only after painstaking analysis would employ more complex constructions than dialogue-oriented trivial literature (Sampson et al., 1988), and in fact some figures exhibit this trend even when comparing results for sentences of comparable length. Comparing these results with the those in Table 3.2, we see that the NEGRA test set in fact obtains the lowest figures, although at 16.7 words its sentences are not the longest on average. This shows that if our grammar contains any bias towards a particular text type, it is certainly not towards the newspaper articles from NEGRA.

Despite these various uncertainties, we feel justified in saying that our approach retrieves the syntax structures of German better than any previously described automatic system.

3.6.4 Error analysis

We have seen that despite competitive accuracy, the syntax analyses that WCDG computes still contain errors; and because of the infeasibility of combinatorial optimization in general, we cannot easily determine whether the search algorithm or the language model was responsible for any particular misanalysis. Nevertheless, we would like to know at least roughly how many mistakes are due to search and to model errors, so that we can direct our efforts for improvement either at the constraints or at the algorithms.

It is easy to find clear instances of *modelling errors* in the experimental results. Consider the analysis of the sentence

"Außerdem erhält der Bauträger ein Darlehen von 579000 Mark."
(The contractor is also loaned DM 579,000.)
(NEGRA, sentence 18602, simplified for exposition)

in Figure 3.3. The word 'von' is misattached so that the interpretation becomes 'the contractor receives a loan from the DM 579,000', which is obviously not the intended meaning. For this short sentence, the optimization problem can be solved exactly, and it turns out that this analysis does in fact carry the highest possible score. The intuitively preferred attachment of 'von' to 'erhält' scores slightly lower because of a general preference for attaching prepositions to verbs rather than nouns. Although helpful in general, the rule fails for this particular example.

Modeling errors occur either because the grammar writer has made a mistake in judging the relative merits of two alternatives, or because the necessary distinction cannot be made with the means at hand. In the first case, the mistake can often be corrected, while in the second case, only a general appraisal is possible of what

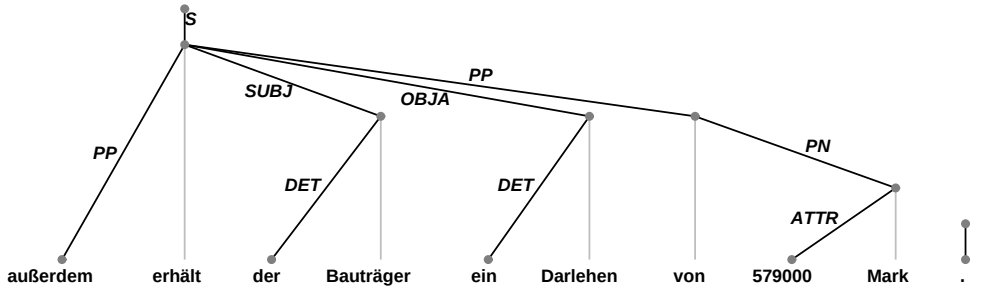

Figure 3.3: A misanalysis due to a modeling error.

kind of distinctions the grammar can or cannot make. For instance, a grammar that employs sortal restrictions could be able to resolve the ambiguity between subjects and objects when one of the readings is semantically unlikely, while a pure surface syntax description could not. A domain-specific grammar could even include rules to distinguish between interpretations when both are equally plausible, but one is known to actually hold in the restricted domain. In this way, even a model known to be imperfect can still merit a limited amount of trust in its predictions, as long as one is aware of the limitations.

On the other hand, *search errors* are more difficult to combat. There is no obvious way in which a search error could be reliably corrected without performing extensive analysis of the typical behaviour of an algorithm on a set of problems. Also, search errors do not occur in particular areas of representation only; when an incomplete solution method computes a suboptimal analysis, it effectively ignores some of the rules that the grammar writer set, and any particular rule might be ignored (although it is likely that rules with higher penalty values will be ignored more often).

For an instance of a search error that leads to a wrong analysis, see the analysis of the sentence

"Fehlbeträge seien allerdings noch in Australien, Südafrika, Japan und bei der Medizintechnik in Großbritannien zu verzeichnen."
(However, deficits were still registered in Australia, South Africa, Japan and in the medical technology department in Great Britain.)
(NEGRA, sentence s19099)

in Figure 3.4. The resulting interpretation is the very strange "But in Australia, deficits are South Africa, Japan, and to register in the medical department in Great Britain." The syntax structure returned by the parser assumes that the name 'Japan' and the verb 'verzeichnen' are co-ordinated, which is a violation of type constraints and does in fact incur a strong penalty according to the grammar. Such an obviously questionable result is only returned if no better analysis could be found at all. Comparison with the preferred reading in Figure 3.5 shows that it contains no fewer than seven structural errors, and none of the intermediate solutions carry a higher score.

Breaking out of this local maximum would therefore require taking a great number of correct heuristic steps, and in this case the transformation algorithm happened to miss at least one of the necessary repairs.[6]

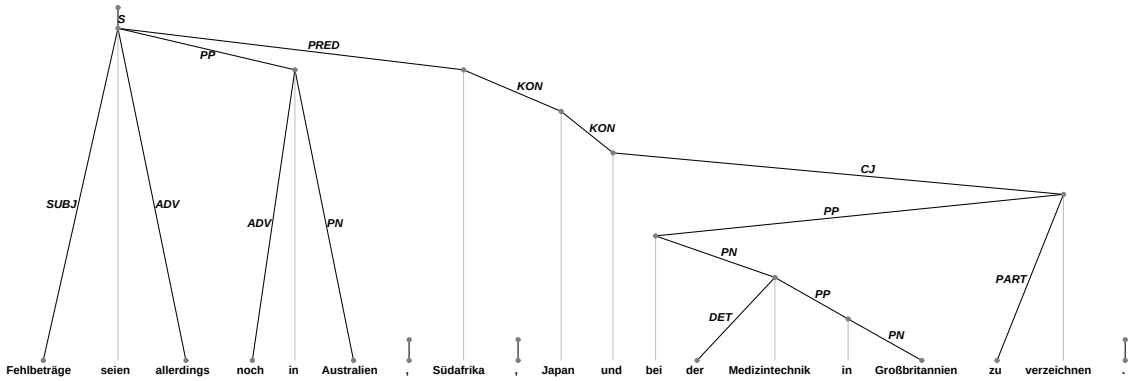

Figure 3.4: A misanalysis due to a search error.

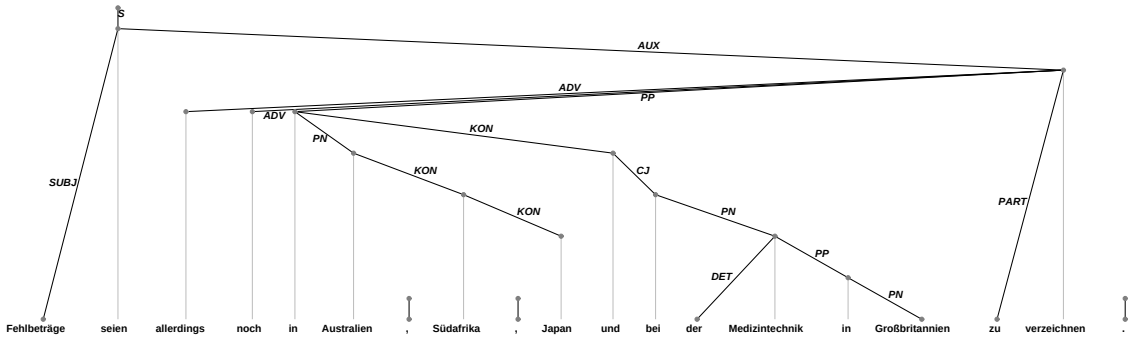

Figure 3.5: The correct analysis of the same sentence.

An important question would now be: how many of the 1622 errors were search errors, and how many were model errors? If the great majority of the errors were model errors, improving the grammar rules would have to be the priority; but if the majority were search errors, better search algorithms would appear more promising. The problem, of course, is that this ratio cannot easily be computed, precisely because the computation is infeasible in general, and can only be performed for short sentences in isolation.

[6]This example is both typical and atypical of the behaviour of the investigated grammar of German. The failure to connect the full verb 'verzeichnen' with its linearly distant regent 'seien' is indeed one of the more common errors made by the parser. On the other hand, such search errors usually occur in the analysis of longer utterances than this; this atypically short example was selected mainly because its tree still fits on one page.

The problem is compounded because search and model errors can occur independently of each other:

- If the search returns an analysis that has a lower score than the annotation, this proves that a search error has occurred. But we cannot know whether a complete search algorithm would have returned the desired analysis, or whether a modelling error might not have been uncovered in this problem instance as well.
- If the result has a higher score than the annotation, this proves that the model contains at least one error, because it does not predict the highest score of all for the desired analysis. But again, there might be a third analysis with an even higher score that the search did not find.
- Likewise, if the search returns the desired analysis, it might still have overlooked another analysis that carries a higher score; in other words, a search error could have canceled out a modelling error.

Nevertheless, by comparing the scores of the parsing results to the scores of the respective annotation, we can at least give lower bounds for the number of sentences which are affected by model errors, and the number of runs in which search errors occurred. It turns out that of the 1,000 parsing runs, 125 resulted in an analysis that carried a lower score than the hand annotation, but 467 found an analysis that scored higher than the annotation. This somewhat corroborates the hypothesis that with the almost unlimited parsing time, search errors do not constitute the predominant error source of errors, and that the model itself has an accuracy closer to the observed 90% than to 100%.

We can also undertake an active search for model errors with the following experiment:

1. Create an instance of the parsing problem for each sentence.
2. Use the hand annotation to find the problem state that corresponds to the intended analysis.
3. Start the transformation algorithm.

If the language model were absolutely perfect, this experiment would result in a dependency accuracy of 100%, since transformation cannot find a wrong analysis that scores better than the hand annotation, and since we start out with the annotated analysis there is no possibility of missing it. In fact, whenever the algorithm achieves an improvement over its starting point this proves that there is an error in the language model with respect to this particular sentence.

If the dependency accuracy is close to 100%, it would speak for a rather reliable language model in that the numeric optimum is usually not very different from the annotation. On the other hand, if the value is lower, perhaps even lower than the accuracy measured previously, this would show that there are a great many misanalyses resulting from model rather than search errors.

It turns out that in many cases, transformation does find an analysis that is numerically better than the hand annotation: the idealized experiment results in a dependency accuracy of 93.8%/92.6%. Remember that this does not necessarily mean that the modelled optimum is off by 6.2% on average; there might always be yet another analysis, even better than the one computed here, that we did not find. This hypothetical analysis might be more or less accurate than the result that was returned; we know only that it is not identical to the annotated analysis, since it has a different score.

All we can strictly prove with this experiment is that for many sentences (430 out of 1,000), the language model is not entirely correct and could be improved. On the other hand, the clear rise in dependency accuracy indicates that the incomplete search was also a source of errors in the previous experiment. This corroborates what earlier investigation of individual parsing errors already suggests: both model errors and search errors contribute measurably to the errors that the WCDG of German makes, and could benefit from systematic improvement. Therefore both new rules and new heuristics for guiding the search among a particular rule set could be useful.

Chapter 4

Enhancing WCDG With Statistical Information

This chapter discusses statistical enhancements to WCDG for three separate purposes:

1. For comparison to related work, we conducted all experiments in Section 3 under the simplifying assumption that the syntactic category of each word is known in advance. This is not only at odds with a proper parser evaluation, since it presupposes some of the information that a parser is supposed to compute, but is also untenable in general: it is only practicable when annotations of each input sentence are in fact available; but a general parsing system must be able to cope with unannotated input. The impact that the assumption has on the accuracy figures in Section 3 is measured. Statistical methods are discussed that allow WCDG to operate on raw text instead.

2. The language model defined by our constraints was found to have systematic weaknesses that cannot easily be mended by writing more constraints. Statistical methods are proposed that help improving the model in key places, thus reducing the number of errors in the numerical model.

3. WCDG learns nothing from previous parsing runs; every sentence is interpreted anew, as if the language were reconstructed from first principles every time it was used. Statistical methods might help to steer the search, thus avoiding some of the search errors previously observed.

4.1 Category information for parsing

4.1.1 Realistic WCDG parsing

The assumption from Section 3 that the syntax category of each word is known in advance is not tenable in practice; it was only introduced in order to make the comparison to previous work on broad-coverage parsers, which is already rather different, slightly more adequate. For realistic evaluation, this assumption must be discarded. Our first task, then, is to repeat the evaluation while allowing all category variants that our lexicon of German covers, and leave it to the parser to determine the most appropriate category for each word.

A note is in order about just how broad this coverage is. Since we want to guarantee robust processing for the parsing of completely unrestricted input, our lexicon has to deal with the fact that its open-class word lists are almost certainly incomplete. In order to ensure complete lexical coverage nonetheless, it contains various *template* entries for open word classes. For instance, these templates allow the parser to interpret unknown words as nouns (if they are capitalized), or as verbs (if they bear one of the characteristic suffixes). But there are also word classes such as `NE` (proper name) or `FM` (foreign-language material), which in theory can be instantiated by every string. For instance, the German pronoun 'ich' belongs to the closed class `PPER` (personal pronoun), and almost invariably the word 'ich' is in fact an instance of this pronoun. However, a parser can never be absolutely certain of this. The string might also denote the Integrated Controller Hub on some integrated circuits — and in fact, it sometimes does[1]. In such instances, the word must properly be tagged as `NE` instead.

Allowing cases like these increases the size of each parsing problem immensely, because it introduces many alternatives that are almost always wrong. If every token can theoretically belong to the class `NE`, then every sentence might in theory be a single long compound name. This would usually result in an almost completely wrong analysis, since the linear structure of noun phrases is very different from the typical tree shape of normal sentences. In the previous experiments, this problem did not occur because the homonyms with implausible categories were excluded from the start.

To counter the increase in categorial ambiguity, we preventively reduce the coverage of the lexicon so that only *completely* unknown words can belong to any open word class, while known words are supposed to be completely known, i.e. not to have any open-class homonyms.[2] For instance, if any of the sentence in the test set *did* contain an instance of 'ich' as a proper noun, this would not be covered. This should avoid the pathological possibilities discussed above while still combining a reasonable

[1] See `http://www.intel.com/design/chipsets/datashts/290655.htm`.

[2] This is accomplished by setting the `templates` variable to a lower level in the cdg program.

coverage of most input, and providing a reasonably fair test for the system on unseen input.

		Accuracy	
Length	Instances	structural	labelled
1 – 10	340	84.6%	79.6%
11 – 20	323	76.2%	72.0%
21 – 30	229	70.0%	65.9%
31 – 40	76	67.9%	64.0%
>40	32	64.0%	59.2%
overall	1,000	72.6%	68.3%

Table 4.1: Results of parsing without known categories.

When parsing the same test set as before under these changed conditions, we obtain the results given in Table 4.1. The degradation in parsing accuracy is dramatic; far from being superior, WCDG now falls behind previous work. As an example of an error that is now made, consider:

"Lamari hatte in der Lockerbie-Frage einen unnachgiebigen Standpunkt vertreten."
(Lamari had taken an intransigent position on the Lockerbie question.)
(NEGRA, sentence 18780)

Figure 4.1 shows the annotated analysis, and below that the analysis computed by the parser in this run. The only difference hinges on the syntactic category of the word 'der'; it should be an article (`ART`), but the parser interpreted it as a substituting pronoun (`PDS`). As a result the structure of the prepositional phrase 'in der Lockerbie-Frage' has become inverted, with 'der' dominating 'Frage' instead of vice versa, and both words have received the wrong regent. This results in a structural (and labelled) accuracy of 8 out of 10 (rather than 10 out of 10, as before).

This error is characteristic of the overall results in several respects. First of all, it causes the accuracy figures to fall drastically; in fact, the drop of 20% corresponds quite accurately to the overall drop from 90.4% to the 72.6% observed now. This is despite the fact that logically, the analysis is essentially unchanged. Note that the decision to place determiners below their associated nouns was taken consciously in order to avoid spurious ambiguity (see Section 3.2.4), but is essentially arbitrary; according to other theories of syntax which espouse 'determiner phrases' rather than 'noun phrases', the lower version should in fact be assumed, and would correspond exactly to the desired interpretation.

Of course, this is not the only type of error that occurs when we compare the results to the previous experiment. Many similar systematic misinterpretations of category can be discerned. Also, we must expect that the much greater average size of the optimization problem has once again increased the noise introduced by search errors. However, the mistaking of a definite article for a demonstrative pronoun is in fact

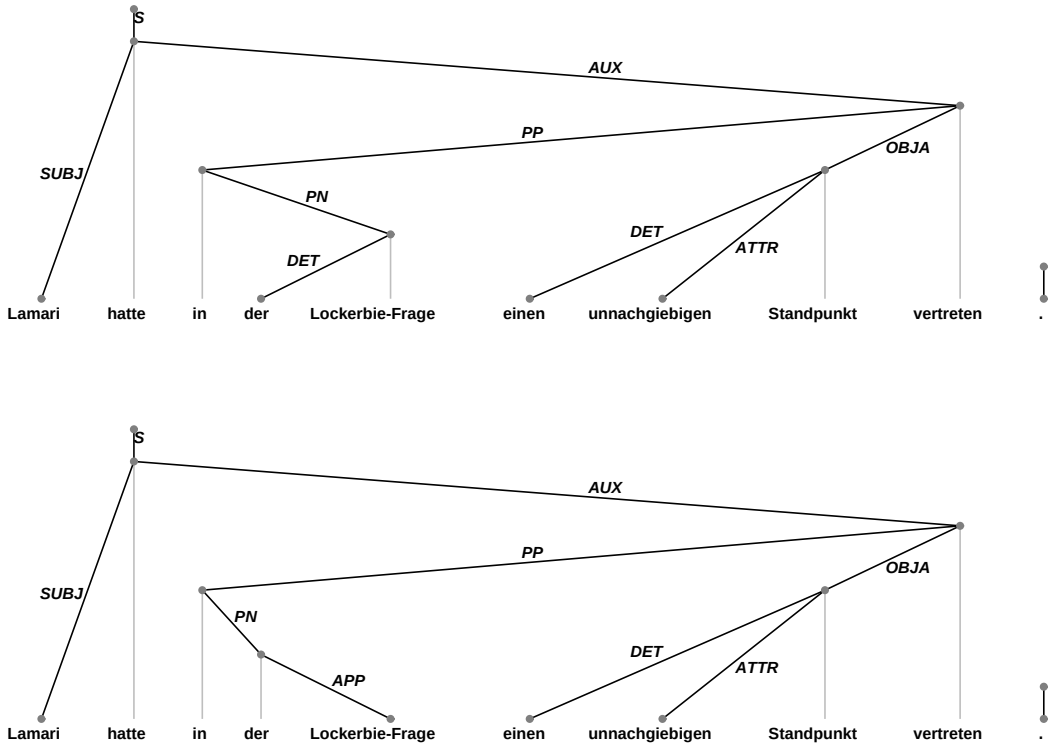

Figure 4.1: A misanalysis due to category ambiguity.

the most common category confusion by far, and is almost universally accompanied by at least two subordination errors. (In contrast, of all the demonstrative pronouns in the test set, only one is mistaken for something else by the parser.)

Finally, this error is characteristic in yet another way: it should be easy to correct. Although it is indeed possible for a demonstrative pronoun to precede a noun directly, this is extremely rare compared to the normal case, an article preceding a noun. In our dependency corpus, the difference is in fact about four orders of magnitude; even a very simple *empirical* model should be able to capture such a strong preference against the pronoun reading. Fortunately, such models have in fact been developed and investigated quite thoroughly. In fact, automatic category classification (*part of speech tagging, POS tagging*) is one of the tasks in natural language processing to which computers have been applied with the most success.

4.1.2 Definition of the part-of-speech tagging task

Formally, the task of a *tagger* consists in mapping the words of a given text to a string of tokens that classify each word as belonging to one or more syntactical categories. Typically, the possible tokens form a comparatively small, closed set that implies only a broad classification (Santorini, 1990; Schiller et al., 1999), but sometimes detailed morphosyntactic information is included in the tags, which leads to many hundred or even thousand possible types (Chanod and Tapanainen, 1995; Hajič, 1998). The mapping need not be bijective; if no sufficient evidence can be found to decide unambiguously for any single class, a tagger may produce more than one possibility (*multi-tagging*) (Voutilainen, 1995). Its performance must then be given as a pair of precision and recall values, rather than simply its error rate.

A typical tagging problem for English would be

> Net was aided by a lower income tax rate.

According to the guidelines set down by Santorini (1990), the correct system response is

> JJ VBD VBN IN DT JJR NN NN NN .

The mapping in this case indicates that “lower”, “tax” and “rate” are respectively an adjective, a noun, and another noun, although all three could be verbs in a different context. Obviously this kind of information is valuable for further analysis, since it can be used to exclude many theoretically possible structures and so reduce the ambiguity considerably.

As with many areas of language analysis, the problems encountered in part-of-speech tagging range from the trivial to the intractable. Compared to other levels of analysis,

a high percentage of decisions can be made with near-certainty even with very basic means such as a simple table lookup, but there are also ambiguities that require much more effort for resolution. In some cases a full syntactic analysis may be required; in others even this may not be sufficient, since the input is genuinely ambiguous.

4.1.3 Previous work on part-of-speech tagging

The history of part-of-speech tagging clearly shows a trend towards data-driven models that continues to this day. A comprehensive dictionary of a natural language often provides enough information to make the majority of all classification decisions in a text, since there is only one possible decision. For instance, every occurrence of 'the' in an English text can be classified by simply looking it up. The ratio of trivial decisions is not quite so favorable when counting *types* rather than *tokens* because the unambiguous words tend to belong to the smaller closed classes of a language. Nevertheless, even in terms of types over 50% of all decisions are typically trivial (DeRose, 1988).

The first published methods of automatic category assignment combined dictionaries of known words with suffix analysis to deal with unknown words (Klein and Simmons, 1963; Greene and Rubin, 1971). However, they already contained some empirical elements as well. When sequences of more than one ambiguous word were encountered, the possible n-tuples were first enumerated completely. Then a corpus of tagged text was searched for those tuples in order to decide which of the candidates should be assumed. The unspoken assumption is, of course, that a category sequence which has occurred once is more likely to recur than one which has not.

Leech et al. (1983) were the first to use both *relative tag frequencies*, which could distinguish between possible but unlikely and actually likely tags for a particular word, and *collocational probabilities*, which judge the probability that two particular categories would appear next to each other. Both refinements are useful for resolving ambiguities that remain after dictionary lookup. For instance, the English word 'meter' can be a noun or a verb, but the noun reading is much more frequent; therefore choosing this reading is preferable if no other evidence is available. But if the preceding word has a unique category, it can offer even stronger evidence either way: for instance, a preceding 'I' strongly suggests the verb reading, while a preceding 'the' suggests the noun reading. While the older context frame rules could only take a decision when all but one sequence were impossible, probabilities made it possible to decide between multiple sequences that are all possible but not equally so.

While Leech et al. (1983) used relative tag frequencies only to manually enforce a dispreference for certain tag/word pairs, the collocational probabilities were automatically computed from a preexisting tagged corpus. The algorithm was able to resolve ambiguous sequences of arbitrary length; however it took exponential time and space

to do so because of its over-complicated probability model, and also employed various other special components and rules that were created entirely manually.

The algorithm VOLSUNGA (DeRose, 1988) finally implemented an almost completely empirical approach, and was written explicitly with this goal. It used dynamic programming in combination with a simpler model of sequence probabilities to reduce time and space requirements to a linear function of input length. At the same time it achieved the same accuracy as earlier results.

Since this publication, both the means of n-gram probabilities and the preference for automatically derived parameters have almost universally been employed. Most of the known methods of machine learning have been applied to the problem, such as (in alphabetical order) Hidden Markov Models (Cutting et al., 1992), loglinear models (Toutanova et al., 2003), Maximum Entropy calculation (Ratnaparkhi, 1996), n-gram counts (Church, 1988; Lezius et al., 1998), neural networks (Schmid, 1994a; Ma et al., 1999), statistical decision trees (Schmid, 1994b), or support vector machines (Giménez and Màrquez, 2003). Altogether the means used for the task of part-of-speech tagging English have proven much more varied than the results: various different probabilistic algorithms have been published that achieve an accuracy of about 97% for arbitrary English text, but none that consistently improves on that number.

However, despite the palpable success of statistical part-of-speech tagging there are still projects that perform category classification in a rule-based paradigm. Voutilainen (1995) describes a system that uses only hand-written declarative rules to analyse English texts. Categorial ambiguity is reduced in several steps: lexicon lookup, morphological analysis, and cyclical application of rules defining contexts that exclude a particular category. Altogether, this system assigns the correct category to 99.7% of all tokens; however, 3–7% of all tokens receive more than one category, i.e., the precision is considerably lower. Further stages of analysis, in particular a deeper syntactic analysis, can lower the categorial ambiguity to almost 1, while accuracy remains significantly above 99%.

This method of part-of-speech tagging has the usual disadvantages of hand-written, rule-based systems: EngCG required years of effort by trained language experts to write, and it cannot easily be transferred to other languages. It is also not generally available, since only the overall approach was published and not the actual rules.[3] Nevertheless this and other results (Chanod and Tapanainen, 1994) indicate that for the most reliable analysis possible, human expert knowledge is still irreplaceable, even in a domain as restricted as this one.

This is particularly true if categorial disambiguation is not undertaken as an aim in itself, but in connection with other processing. The most common use of category symbols is certainly as an aid to full syntactic analysis. Many methods of parsing rely on them to work, e.g. to decide whether or not a particular production rule

[3]The authors now offer their language processing systems commercially.

can apply to a certain pair of words (de Marcken, 1990; Weischedel et al., 1993). Other systems are made practicable only by prior part-of-speech tagging, or even dispense with the words themselves and *only* use category symbols (Charniak et al., 1996). This is often the case when a broad-coverage parser is to be applied to long input sequences: Langer (2001) reports several dozen possible analyses for a German sentence composed of only twelve words, and millions of analyses are not uncommon for realistic sentences. In these cases, part-of-speech tagging can be employed as a filtering step that marks at least some of the wrong analyses as unlikely. These analyses can then be either be suppressed or dispreferred if the need arises.

The combined advantages of efficiency, high accuracy, and immediate benefit through disambiguation were such that part-of-speech preprocessing very quickly became a standard component of automatic parsing: while Charniak et al. (1996) stated that "it is still uncommon actually to read of a tagger used with a parser", only a few years later parsers *without* a part-of-speech tagger were uncommon enough that the fact had to be mentioned explicitly (Braun, 2003).

Of course, it is only helpful to employ a dedicated part-of-speech tagger in a general language analysis system if this actually improves the overall performance, e.g. with respect to efficiency or accuracy. If a parsing system were able to parse untagged input reliably, no tagging component would be needed, since category assignment is a proper subtask of the one solved by a parser: most definitions of syntax trees encompass category and morphosyntactic features. But generally applicable parsers are usually not able to guarantee perfect retrieval of all word categories, any more than they can retrieve syntax structures perfectly, therefore external helper programs that exceed the parser's ability in some restricted area can often be actually found. In particular, parsers can usually profit from the help that part-of-speech tagging provides.

Several reasons can be given for this. Concentrating on a subtask of language analysis often allows more effort to be expended on it; for instance, most statistical part-of-speech taggers will acquire preferences not only for sequences of categories but of category/token pairs, while full parsers are not always easily lexicalized. Also, some parsers fail to produce any output for input that does not conform to their model of language, while the typical part-of-speech tagger will simply assign the most probable category sequence, no matter how improbable it may be in absolute terms.

Although not all of these arguments apply to our case (WCDG itself is quite robust in the sense that no input is rejected outright), the same tendency can be observed. If we measure the accuracy of the last parsing experiment only in terms of category assignment, it turns out that only 89% of all words are classified correctly as a by-effect of parsing. This is far below what dedicated algorithms can achieve, so that importing more reliable category predictions could yield a benefit to the overall system.

4.1.4 Part-of-speech tagging for WCDG

The preceding examples suggest quite forcefully our WCDG of German would benefit from a component for category classification that is *external* to the optimization problem itself. Although the typical category ambiguities in German text are different from those in English (see Section 4.1.6), the accuracy achievable by part-of-speech taggers seems quite similar for both languages; Brants (2000b) reports an accuracy of 96.7% for the NEGRA treebank.

The TnT program that Brants described is not only accurate and very fast, but it is also freely available for research purposes. It also uses precisely the same set of syntactic tags as our grammar and ships with a precompiled model of German categories; if we switched tag sets it would also allow retraining. If the program is run with its own model of German on the example sentence of the previous section, its output is the following:

```
Lamari          NE    1
hatte           VAFIN 1
in              APPR  1
der             ART   1
Lockerbie-Frage NN    1
einen           ART   1
unnachgiebigen  ADJA  1
Standpunkt      NN    1
vertreten       VVPP  0.761 VVINF 0.169 VVFIN 0.062 ADJD 7.938e-03
.               $.    1
```

All ten words are in fact correctly classified; in particular, the word 'der', which might conceivably be an article or a demonstrative or relative pronoun, is resolved to be unambiguously an article. Only the word 'vertreten', which might be a past participle, an infinitive, a finite verb, or a deverbal adjective, is in fact classified as all four, although with different degrees of certainty. TnT is configured so that it emits all possible tags which are found in a beam search with a constant beam width; in this case all four probabilities were comparatively close to each other, although the correct reading `VVPP` is still the most probable.

Obviously this kind of information could have been used to avoid the parsing error for this sentence, if the constraint grammar had access to it. With a view to further applications of the same principle, a rather general system of *predictor* components was added to the implementation that allows arbitrary programs to be called when a sentence is analyzed. It comprises the following:

- A new command 'predict' is added to the CDG command language that registers an external predictor program (or turns it off again, if desired). Whenever

a sentence is to be analysed, this program is called on the string of words that constitute the sentence; the output of the program must consist of tab-separated key/value pairs for each word in the sentence. Each word in the parsing problem is then decorated with the relevant key/value pairs.

For instance, TnT with its recommended parameters could be registered as a predictor with the command line

```
cdgp> predict tagging 'tnt -v0 -z200 models/negra.tnt -'
```

- A new operator 'predict' is added to the constraint language, which returns these predictions. Since there can be more than one predictor in operation at a time, it is a ternary operator; for instance, within a constraint formula the term

  ```
  predict(X@id, P, K)
  ```

  would return the value that the program registered under the name 'P' returned for the word in question under the key 'K'. This operator can then be used in constraints that allow or forbid constructions based on the output of external programs. When no value has been predicted for a particular key, 0 is returned.

It remains to be decided how to combine the predictions made by the n-gram tagger with the penalties on constraints in the WCDG. The obvious thing to postulate about tagger scores is to demand that they should be high; this suggests writing a constraint with a variable penalty that depends on the output of the tagger:

```
{X:SYN} : tagger : POS : [ predict(X@id, POS, X@cat) ] :
  predict(X@id, POS, X@cat) = 1.0;
```

This constraint reads the prediction made by the category tagger (registered under the name `POS`) for the category that the dependent of the current syntax edge in fact has (`X@cat`). If the value is 1, the constraint is satisfied and has no effect. If the value is lower than 1, the constraint fails, and the dependency edge receives a penalty that corresponds to the prediction. Note that when no prediction has been made for a particular key, the `predict()` function returns 0. This means that category variants which were not mentioned by TnT at all are in fact excluded totally, since the 'tagger' constraint equates to a hard constraint in such cases. Thus, the wrong analysis of 'der' as a pronoun cannot be selected.

A minor point to consider here is that TnT returns *probabilities* while WCDG uses *penalties*. The predictions that are made for a word must sum to 1, so when more than one prediction is made, they must all be lower than 1. But in WCDG, any

score below 1 indicates some sort of error; when TnT is uncertain about a category, *all* possibilities would be marked as wrong (although not to the same degree). This would be inappropriate, since the values express *uncertainty* rather than *incorrectness*. Although this does not change the nature of the optimization problem, it is still undesirable from the grammar writer's point of view for several reasons:

- Since one of the category variants *must* be correct, there is no point in expressing a preference against it. While it is by no means certain that the category with the highest value according to TnT is indeed correct, this is at least possible. On the other hand, assuming that all categories are wrong cannot possibly be true.
- Constraint violations drive the transformation process; therefore a tagger constraint that cannot be satisfied at all would cause repair attempts that cannot succeed, and merely take up time for no gain.
- Generally, a problem that has no solution with a high score is somewhat more difficult to analyse than one that does (Foth, 1999b), because the probability that cautious pruning can reduce the size of the problem rises with the score of the optimal analysis.

Therefore, rather than using the output of TnT directly as predictions, we normalize it by dividing all values by the highest value. This leaves the relative preferences among alternative categories unchanged, but avoids any penalty in the most common case that the most probable category is indeed selected.

		Accuracy	
Length	Instances	structural	labelled
1 – 10	340	94.3%	92.4%
11 – 20	323	90.8%	88.8%
21 – 30	229	88.6%	86.8%
31 – 40	76	86.7%	85.0%
>40	32	82.4%	80.6%
overall	1,000	89.0%	87.1%

Table 4.2: Parsing results with the tagger constraint.

Table 4.2 shows the results of the parsing with the additional tagger constraint. In comparison with Table 4.1 and Table 3.2, we see that the statistical tagging component improves parsing accuracy almost to the level of the initial experiment where category ambiguity was artificially excluded.

4.1.5 Previous work on part-of-speech tagging for WCDG

Wide-coverage WCDG parsing as discussed here clearly falls into the class of approaches that are made feasible only by reliable part-of-speech tagging. All previous work in WCDG dealt with much more restricted subsets of German. For instance, the system described in Menzel and Schröder (1998) did not cover relative or demonstrative pronouns at all, and therefore did not have to distinguish them from articles.

The integration of part-of-speech tagging into WCDG as a prediction component was first described by Foth and Hagenström (2002). The first experiment described there used a grammar fragment tailored to simplified utterances from Verbmobil dialogue This corpus consisted predominantly of short exchanges such as the following:

"Wie wäre es denn im Juli?" — "Anfang Juli hätte ich noch Zeit."
("How about July, then?" — "I still have time at the beginning of July.")
(Verbmobil, n009k001-2)

This grammar achieved a structural dependency accuracy of 96.3% on its test set of 222 sentences. The addition of a tagging constraint did not improve this value, but it reduced the time spent by the transformation algorithm by up to 50% (for the largest problems) while achieving similar accuracy. Various variants of tagger integration were investigated; running TnT in its multi-tagging mode and normalizing its predictions as described previously achieved slightly better results than gradating its influence through a more complicated penalty formula.

The second experiment was conducted on a very early version of the grammar of German described here, which already covered a much wider variety of syntactic phenomena and was also able to cover unknown names, nouns and adjectives. In this case, reliable part-of-speech information proved to be even more valuable, approximately halving the error rate for structural attachment.

While the utility of part-of-speech classification judgements is apparent, it is not immediately obvious why employing an external utility is the best way to obtain them. Since WCDG in principle allows arbitrary conditions to be expressed, could not the same result be obtained by writing CDG constraints on the categories of adjacent words? Although some of the more obvious regularities could probably be captured, this is not an attractive option for several reasons:

1. WCDG constraints are capable of relating at most two dependency edges to each other, but category disambiguation frequently involves considering at least three adjacent words in an input string. Even algorithms that base their calculation on trigram possibilities take more context into account indirectly. This is not possible in WCDG without introducing completely new operators.

2. Perfectly usable part-of-speech taggers are already available, and it is not certain that re-implementing them as WCDG formulas would achieve a comparable accuracy. In fact, since all constraints are evaluated in a single dimension,

this would make it impossible to measure the accuracy of part-of-speech tagging independently from syntax analysis.

3. Calling an external utility for obtaining category probabilities ensures that the necessary computations are performed only once, while constraint formulas can be evaluated many times during a disambiguation process.

4.1.6 POS tagging as an error source

The positive effect of external part-of-speech tagging on parsing has been confirmed various times. The work reviewed in Section 4.1.4 showed that this also applies to WCDG: the ambiguity reduction provided by a tagger is helpful enough that it outweighs the disadvantage incurred by individual tagging errors, even if these lead to a worse result for some parsing runs. Nevertheless, its effect is usually better the better it is at its own task, since errors in the first step can directly cause processing failures at later stages. It is not unusual for authors of broad-coverage grammars to state that they are the single largest source of misanalyses in their system (Bangalore et al., 1996; Bleam et al., 2001; Dubey, 2005).

Foth and Hagenström (2002) also noted the problem of tagger errors. It was shown for an example analysis that WCDG can accommodate erroneous tagging decisions if the correct category is not excluded altogether; although the annotated analysis was incorrectly penalized, it was still the best analysis numerically, although it became more difficult for the solution algorithm to find. However, no systematic investigation into all occurring errors was made.

It is easy to find individual instances of parsing errors that can be explained through tagger errors. Consider the following example:

"Als Miete werden angeblich nur etwas mehr als 60 Mark pro Quadratmeter und Monat berappt."
(The rent is said to be just slightly above 60 DM per square meter and month.)
(NEGRA, sentence 19454)

The preferred analysis is shown in Figure 4.3. But in our experiment, the part-of-speech tagger classifies the first word 'als' as a subordinating conjunction (`KOUS`), while actually it should be tagged as a comparator (`KOKOM`). Figure 4.2 shows the resulting analysis. As a result of the tagging error, the first three words of the sentence are incorrectly assumed to form a subordinated clause (even though there is no main clause to subordinate it to). The object of the comparison, the word 'Miete', cannot attach to its true regent, and has been parsed as the subject. In turn, the true subject 'Mark' is ousted from this position, and has to attach as a right extraposition. Also, the full verb 'berappt' cannot form the correct auxiliary phrase with the verb 'werden', because this would violate the canonical word order for German subclauses; instead it subordinates as an adverbial modifier.

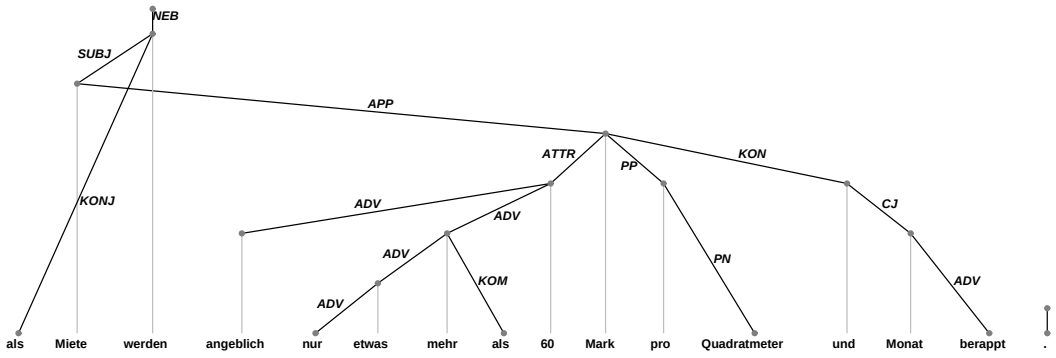

Figure 4.2: A misanalysis due to a tagging error.

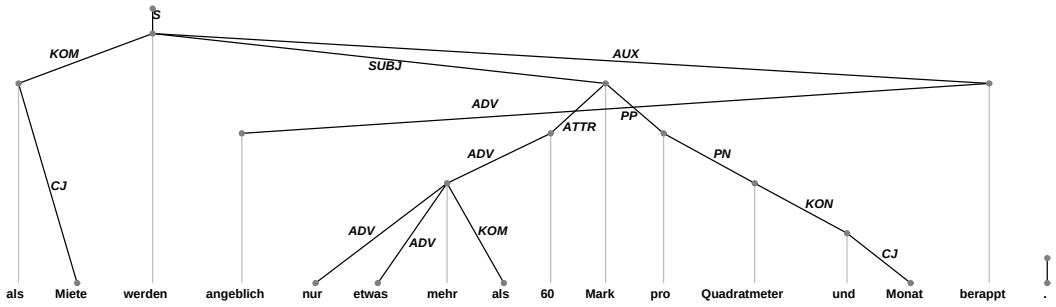

Figure 4.3: The correct analysis of the previous sentence.

Altogether, the error has caused six words to be attached to the wrong regent (note that the mis-tagged word 'als' itself is attached correctly, although not with the correct label). This confirms the plausible assumption that the beneficial effect of including statistical information can sometimes turn into a disadvantage: although tagging errors are perhaps not the most common source of errors in our system, they can cause disproportional damage in individual cases.

In the terms of Section 3.6, it is not clear whether such effects should be viewed as search errors or model errors. The intention of part-of-speech preprocessing was to guide the parser through an overly large search space, in order that it should take less effort to find the solution that would have been found anyway, i.e. to avoid search errors arising from the incomplete algorithms. But in order to achieve this effect, the tagger scores change the very evaluation function that we wanted to optimize, so that technically, errors committed because of wrong part-of-speech tags are modelling errors. It is perhaps best to classify them as a separate category of imperfection: errors due to a *failure of heuristics*, the dark side, as it were, of the trade-off that we make in including statistical information sources into our handwritten parser.

Since it is characteristic of heuristics that they occasionally fail,[4] a residue of such errors must be expected. Nevertheless we should ask at least the following questions:

1. What proportion of parse errors is due to failed heuristics?
2. Can we reduce these errors without losing the benefit of statistical information?

4.1.7 Correction of part-of-speech tagging errors

The error discussed in Section 4.1.6 is characteristic of the failures of statistical components in that it is clearly due to the simplistic assumptions used in creating them. In the case of TnT, the assumption is that local context is always sufficient to disambiguate word classes. The sentence concerned contains strong indicators against the classification of 'als' as a subordinating conjunction: it contains no second verb, and the first verb is not marked with a comma. However, apparently the local context was not large enough to capture them. This kind of error cannot easily be avoided with n-gram models, since there is no limit on how far the second verb that justifies assuming a subclause might be removed from the conjunction; a global perspective on the entire sentence would be required to detect both indicators. But the number of parameters required for an n-gram similarity model rises so quickly with n that this sort of global model is infeasible to compute (and would require immense amounts of training material).

[4] "Heuristics are bug-ridden by definition. If they didn't have bugs, then they'd be algorithms."
— anonymous

It has previously been suggested to take active measures in order to correct errors that n-gram taggers typically make. For instance, Briscoe and Carroll (1995) describe a combination of PCFG and unification grammar that is ultimately intended to achieve an accuracy comparable to that of part-of-speech taggers. Their grammar "incorporates some rules specifically designed to overcome limitations or idiosyncrasies of the tagging process"; in other words, it contains workarounds for known tagger bugs.

While these authors incorporate known weaknesses of part-of-speech tagging directly into their language model and so create a rather tight coupling of part-of-speech tagging and parsing, it seems more natural to directly improve part-of-speech tagging for the sake of a parser, rather than change one's parser to suit an ideosyncratic part-of-speech tagger. Wauschkuhn (1995) describes the addition of statistical part-of-speech tagging to a partial parser of German and discusses its effect on coverage and ambiguity reduction. In addition to problem-specific aspects, he also proposes various more general steps that could improve the collaboration between part-of-speech tagging and partial parsing. His first recommendation is that "frequently repeated tag errors must be recognized", since his tagger has an error rate of 4% and often causes even partial parsing to fail. An obvious solution would be to repeat failed parsing runs without the part-of-speech tagger. However, he also recommends that either the tagger should be improved with respect to these cases, or multi-tagging should be employed for known difficult decisions. The tagger could even be restricted to predicting only those parts of speech that actually can be decided on the basis of the considered context.

Schmid (1996) discusses various measures to improve tagging accuracy for German, which differs considerably from the corresponding in English (a productive morphology increases the number of model parameters considerably, and much less training material is available). After evaluating the improvements, he notes that 20% of all remaining errors are confusions between finite and infinite verbs. Special postprocessing is suggested for filtering out such errors.

Both these authors note that even though the overall problem is comparable, the distribution of errors encountered in German part-of-speech tagging is very different from that for English. For instance, German capitalization almost entirely eliminates ambiguity between nouns and verbs. On the other hand, certain types of distinction that are very important for German sentence macro-structure are difficult to get right for automatic taggers. For instance, infinitives most often occur when a finite auxiliary verb precedes, but the distance between the two is usually to great for an n-gram tagger to learn. At the same time, the resulting wrong predictions are serious impediments for any parser that relies on them, since the correct type of verb phrase cannot be built.

It seems worthwhile, therefore, to actually perform postprocessing such as the one advocated by Schmid. For instance, the knowledge that a finite verb form is more probable when a subordinating conjunction precedes it can be cast into a simple rule

that looks for such conjunctions and then changes the final 'infinite' tag to 'finite' in the hope that this will be correct.

In order to have a larger base of examples than the 1,000 example sentences parsed in the previous experiment, we take 10,000 sentences of the 'heiseticker' corpus that have manually created annotations, and note the manually disambiguated part-of-speech tags in them. The unmodified model of TnT for German is now applied to these sentences and then compared to the manually annotated part-of-speech tags. Since this corpus is markedly different from TnT's training material, we would expect a degradation against the previously reported 96.7% accuracy, and in fact it does occur: only 94.7% of all words are assigned the correct tag.

actual:predicted	instances	actual:predicted	instances
FM:NN	1359	FM:ADJD	98
FM:NE	1296	FM:CARD	94
NN:NE	971	VVPP:VVFIN	73
NE:NN	687	ADJA:VVFIN	70
KOKOM:APPR	454	ADJD:VVPP	63
VVINF:VVFIN	354	CARD:NE	60
VVFIN:VVINF	247	ART:PRELS	59
PIS:ADV	164	PIDAT:PIAT	56
PRELS:ART	163	NE:ADJD	54
ADJD:ADV	151	KOKOM:PWAV	53
VVFIN:VVPP	146	FM:PPER	50
KON:ADV	143	FM:XY	50
ADV:APPR	134	NE:ADJA	50
NN:ADJA	123	NN:ADJD	48
VAFIN:VAINF	117	ADV:PIAT	43
FM:ADJA	107	VVINF:VVPP	43
ADJA:NN	100	VMFIN:VMINF	41

Table 4.3: The most common errors committed by TnT on 175853 words of German.

What is more interesting is the distribution of error types that we encounter (see Table 4.3). First of all, it differs clearly from that observed in Section 4.1.1. There, the most common error by far was to mistake a definite article for a demonstrative pronoun, since all of their forms are identical in German. TnT only rarely makes this mistake, since the immediate context quite strongly indicates the correct choice for this distinction: articles very typically precede nouns or adjectives, while substituting pronouns do not. However, definite articles are also systematically homonymous with relative pronouns, whose typical context is much more similar, and consequently the two classes are confused much more often. It is illuminating to study examples of those rare cases where TnT does mistake an article for demonstrative or vice versa:

"Der Adapter ersetzt **das** bisher in die Konsole integrierte Modem."
(This adapter replaces the modem previously built into the console.)
(heiseticker, item 10083)

"Für die verladende Wirtschaft bedeutet **das** Kostensenkungen, für die Reeder Zugang zu neuer Ladung und bessere Auslastung der Kapazitäten", sagt Giesenkirchen."
(This means cost savings for the shipping industry and access to new types of freight and better capacity utilization for ship owners, says Giesenkirchen.)
(heiseticker, item 10976)

Both of these mispredicted (highlighted) words occur in an untypical context (an article far removed from its noun, a substituting pronoun preceding a noun) and require detailed syntax analysis to get correct. This demonstrates that part-of-speech tagging is of value for WCDG primarily because it contributes knowledge from a primarily sequence-oriented paradigm, and so can achieve synergy with a structural analyzer that is quite unconcerned by mere adjacency.

Some of the prevalent types of error that do occur are clearly due to systematic differences between the training and test sets. For instance, foreign-language phrases are much more common in technical articles than in general newspaper copy. Expressed in numbers, only 0.1% of all words in the NEGRA corpus bear the category `FM`, but 0.2% of the words in the 'heiseticker' corpus do. This leads to a lower probability of predicting this category, particularly for unknown words; and indeed, many instances of `FM` are not recognized by TnT. In fact the four most common error classes are all confusions between the categories `FM` (foreign-language material), `NN` (normal noun) and `NE` (proper noun). These categories are very difficult to distinguish[5] without detailed morphological tests that TnT does not make, especially if the word form has never been encountered in training. However, none of these errors is likely to cause much trouble for a parser, since all of these categories perform much the same duties in German[6]. (Brants, 2000a) reports that this type of error is also the one that human judges make most often when annotating sentences for the NEGRA corpus.

The next most common error, `KOKOM` mistagged as `ADV`, results from a modelling difference between the 1995 Version of the STTS (which tags the comparator 'als' as `KOKOM`) and the training material of TnT's training corpus (which apparently assumed it to be `APPR`), and can indeed be trivially corrected.

The next two errors show the verb confusion mentioned earlier, which results from the fact that every German infinitive has a homonym among the plural forms. This type of error is crucially different from the previous one, since finite and infinite verbs do *not* fulfill the same syntactic roles: while a finite verb almost always heads a full

[5]Note that German capitalizes both proper and normal nouns.
[6]The main benefit of distinguishing `FM` from `NN` at all is that `FM` tokens might on occasion perform other functions than that of an NP, e.g. as an adverbial: 'Sie verbreitete die Nachricht *en passant*.' `NE` tokens differ from `NN` tokens in that they often occur without a determiner.

sentence, an infinite verb implies the presence of at least one other superordinated verb in the sentence; also, the possible type and order of their dependents is different. Interestingly, this sort of error is not mentioned by Brants (2000a): the different environments in which finite and infinite verbs occur appear to be so distinctive that human annotators have no trouble distinguishing them. An n-gram model, on the other hand, cannot capture the necessary conditions well. This error, then, is an example of a real limitation of the tagger, rather than just a reflection of an insufficiently detailed annotation specification. It is also likely to cause much more problems for parsing, since one misinterpretation might lead to another, as an entire clause is mis-analysed.

The prediction mechanism introduced in Section 4.1.4 is general enough that arbitrary programs can be specified as information source. We can therefore replace TnT itself with a helper application that first passes its input along to TnT, and then performs postprocessing on the output. An experiment was conducted as follows:

- The sequence of tokens was approximately segmented into 'clauses' by regarding all instances of sentence-internal punctuation as clause boundaries.
- Transformation rules were then applied on the string of tokens that detect common indicators of tagging errors. For instance, the occurrence of two finite verbs in the same clause is a very strong indicator that at least one of them has been mis-tagged. Conversely, an infinitive occurring with no finite verb but preceded by a subordinating conjunction is more likely to be a finite verb.
- Rules could investigate either just the current clause or the entire string; e.g. the conjunction 'noch' can be distinguished from the adverb 'noch' by the fact that it must be preceded by 'weder', no matter how far back.
- Altogether, 40 sequentially applied rules were used. Examples of simple and more complicated rules are given in Table 4.4.

As Table 4.5 shows, a distinction such as that between `VVFIN` and `VVINF` can indeed be improved by such relatively simple postprocessing; the number of errors of this type is reduced by over 70%. More often than not, the remaining errors instances are due to the inaccurate determination of clause boundaries (since not every punctuation character actually starts a new clause) and secondary errors (for instance, if the subordinating conjunction that would have triggered a correction was itself mis-tagged). Only the great minority of cases really are more complicated than the rules assume. Since our postprocessing does not address the most numerous errors, but only those that were suspected to be particularly harmful to syntax analysis, the overall tagging performance does not rise by the same amount; in fact it only increases to 95.7%, i.e., the error rate is reduced by only 17%.

Error type	Correction rule
KOKOM:ADV	If 'als' is tagged 'ADV', change the tag to 'KOKOM'
ADV:APPR	Always tag the sequence 'nach wie vor' as 'ADV KOKOM ADV'
KON:ADV	If 'noch' is tagged 'ADV', and 'weder' precedes, change the tag to 'KON'
PRELS:ART	If a phrase starts with a definite article ('ART'), and ends in a verb-final auxiliary phrase, change the 'ART' to 'PRELS'
VVFIN:VVINF	If a full verb infinitive occurs in a phrase that does not contain either an auxiliary verb or an immediately preceding infinitive particle, change 'VVINF' to 'VVFIN'

Table 4.4: Example postprocessing rules for POS tagging of German.

actual:predicted	instances	actual:predicted	instances
FM:NN	1359	ADV:PIS	62
FM:NE	1296	CARD:NE	60
NN:NE	971	PIDAT:PIAT	56
NE:NN	687	NE:ADJD	54
ADJD:ADV	149	KOKOM:PWAV	52
PRELS:ART	125	FM:PPER	50
NN:ADJA	123	FM:XY	50
VVINF:VVFIN	109	NE:ADJA	50
FM:ADJA	107	NN:ADJD	48
ADJA:NN	100	ADV:APPR	45
FM:ADJD	98	ADV:PIAT	43
FM:CARD	94	VVINF:VVPP	41
VVFIN:VVPP	84	VVPP:VVFIN	38
VVFIN:VVINF	73	FM:ADV	36
APPR:ADV	68	PTKVZ:APZR	34
ART:PRELS	68	KOUI:APPR	33
ADJD:VVPP	64	FM:PRF	30

Table 4.5: The most common errors on the same corpus after postprocessing.

4.1.8 Experiments

Although encouraging, this experiment does not prove very much taken by itself: it is not surprising that a handcrafted correction rule can indeed correct the error that inspired it. The rise in overall tagging accuracy merely shows that the correction rules do not inadvertently introduce more errors than they correct. In other words, they are *specific* enough that they apply to the observed examples. For a meaningful evaluation, we should also test whether the rules are also *general* enough that they are useful in other contexts.

A more useful evaluation is to measure the effect of our postprocessing on the data used in Section 3.6. When TnT is applied to these 1,000 sentences, it achieves an accuracy of 97.2%, which is significantly higher than its result on the previous corpus, and indeed higher than its own documentation reports. This is to be expected, since we are now applying it to a section of its own training set. However, the distribution of serious errors is quite similar to what we measured previously on news ticker sentences (see Figure 4.6).

Note that we use a version of the NEGRA corpus that was systematically transformed by us into a dependency treebank that conforms to our dependency model of German. For instance, this includes changing the tag `APPR` to `KOKOM` when applied to the word 'wie', since this is what our lexicon says. This is why the error KOKOM:APPR reappears in this table, although it would not count as an 'error' in comparison with the original NEGRA corpus. The next most common serious error is again the confusion between finite and infinite verbs, which occurs 19 and 12 times, respectively. Postprocessing reduces these numbers to 9 and 3. Altogether, the postprocessing increases the tagging accuracy to 97.7%; this is a smaller improvement than measured in the previous experiment, since the basis of comparison is higher.

The most important experiment, of course, is to find out how much the improvement of the part-of-speech tagger increases the parsing accuracy. Table 4.8 collates the earlier experiments with two new configurations. Experiment 1 repeats the overall results of Table 4.1 (no part-of-speech preprocessing at all); experiment 2 corresponds to Table 4.2 (plain TnT). The new experiment 3 employs TnT with postprocessing; the difference between the results of experiments 2 and 3 should clarify whether part-of-speech tagging errors really are a problem. Experiment 4 gives a soft upper bound for what experiment 3 might have achieved: it uses an oracle function that looks up the annotated category for each word and then manipulates the output of TnT so that additional predictions are inserted with the value 1 where they are missing; in other words, it simulates what would have happened if TnT were imperfect but our postprocessing perfect (i.e. if it added the correct tag in every case). Finally, experiment 5 repeats the unrealistic results of Table 3.2, where categories were known in advance, as an absolute upper bound to the utility of part-of-speech tagging. For each experiment the structural and labeled accuracy is shown, as well as the reduction in the attachment error rate measured against the baseline experiment 1.

actual:predicted	instances	actual:predicted	instances
NN:NE	45	PIS:ADV	7
KOKOM:APPR	38	PWS:PRELS	7
FM:XY	36	ART:PRELS	5
ADJD:ADV	29	VMINF:VMFIN	5
FM:NE	28	PIAT:PIDAT	4
NE:NN	20	PIS:ART	4
VVFIN:VVINF	19	PTKVZ:ADJD	4
ADV:APPR	16	VVINF:VVPP	4
KON:ADV	12	$.:$(	3
VVINF:VVFIN	12	APPR:KOUS	3
ADJD:VVPP	11	APZR:APPR	3
PRELS:ART	11	KOKOM:KOUS	3
ADJA:NN	10	KOUI:APPR	3
ADV:PIAT	8	KOUS:APPR	3
VAFIN:VAINF	8	NN:NNE	3
VVFIN:VVPP	8	PIDAT:PIAT	3

Table 4.6: The most common tagger errors on 1,000 sentences from the NEGRA corpus.

actual:predicted	instances	actual:predicted	instances
NN:NE	46	KOUI:APPR	4
FM:XY	36	PIAT:PIDAT	4
ADJD:ADV	29	PTKVZ:ADJD	4
FM:NE	28	VVFIN:VVPP	4
NE:NN	20	VVINF:VVPP	4
ADJA:NN	10	APPR:ADV	3
ADJD:VVPP	10	APPR:KOUS	3
VVFIN:VVINF	9	APZR:APPR	3
ADV:PIAT	8	KOUS:KOKOM	3
PIS:ADV	7	NN:FM	3
PRELS:ART	7	NN:NNE	3
PWS:PRELS	7	PIDAT:PIAT	3
ART:PRELS	6	PTKVZ:APZR	3
PIDAT:PIS	6	VMINF:VMFIN	3
ADV:APPR	5	VVINF:VVFIN	3
NN:ADJA	5	ADJA:PIS	2

Table 4.7: Effect of postprocessing on the same problem.

Experiment	Tagger	accuracy	error rate reduction
1	none	72.6%/68.3%	—
2	TnT	89.0%/87.1%	59.9%/59.3%
3	TnT + corrections	89.3%/87.5%	60.9%/60.6%
4	TnT + oracle	89.7%/88.0%	62.4%/62.1%
5	oracle only	90.4%/88.8%	65.0%/64.7%

Table 4.8: Effect of POS preprocessing on parsing the test set.

The first conclusion is that the primary benefit of category prediction is having it at all, while the minutiae of various refinements matter much less. Comparing experiments 1 and 2 shows that using TnT to predict syntactic categories reduces the error rate for syntax attachments by 60%. In comparison, second-guessing its output has only improved this figure to 61%.

To put this figure into perspective, one must compare experiment 1 with experiments 4 and 5. Only the configuration of experiment 4 is remotely realistic (the recall of a tagger can be raised through multi-tagging while retaining an adequate precision, although hardly to 100%); and this experiment shows a reduction of 62% in the rate of attachment errors. This means that our postprocessing rules achieve about half of what would theoretically be possible, and therefore, further work on rule-based tagging might yield some improvement.

On the other hand, many instances in which the rules fail to correct a tagging error cannot be tackled at all with the simple transformation patterns that were used. At the same time, the overall effect of tagger postprocessing parsing accuracy, while significant, is not very large. In fact, even the idealized experiment 5 only achieves an error rate reduction of 65%. This indicates that part-of-speech tagging can be only one component in the search for a truly reliable broad-coverage parser, and we have already nearly exhausted its potential.

We can now answer the questions asked on page 103:

1. At most 0.7% of all dependencies are misattached because of wrong predictions by the part-of-speech tagger.

2. We can reduce this figure to 0.4% with simple rules, and still benefit from the advantage of ambiguity reduction through category predictions.

In other words, tagging errors as discussed in Section 4.1.6 are certainly disruptive in the individual case, but they are not a major source of parsing errors; their influence is small both in terms of absolute dependency precision, and when compared with the overall benefits of tagging. In fact, part-of-speech tagging must be viewed as an 'enabling' technology, that is, one that moves broad-coverage parsing with WCDG from the theoretically possible to the practical. All further experiments will employ the hybrid tagger in addition to the components discussed there.

4.1.9 Comparison

The idea of part-of-speech tagging by successive transformation rules is not original; in fact, a tagger has previously been described that uses *only* transformation rules rather than a two-step approach (Brill, 1995). In this implementation, the rules were learnt automatically by providing the general format of possible pattern-action rules, and always choosing the one that improves tagging accuracy on the training corpus most. The major difference to our experiment is that Brill's rules are learnt entirely automatically from corpus data, and therefore require no human intervention to write. However, this also means that every conceivable rule of a valid form has to be tried out in every step, which leads to very large search spaces even for simple rules. Therefore, the format of Brill's possible rules is much more restricted than in our approach which can condition on any feature of the input. Even so, a purely transformation-based tagger requires very long training times.[7] The advantage of this approach is that no human expertise is necessary at all except for defining the format of possible rules; therefore it can carry over to any language for which there is a large training corpus.

The advantage of the hybrid system described here is that a relatively small number of rules that directly deal with errors actually observed can measurably improve tagging performance over a purely statistical method. Although this means additional effort over simply reusing existing taggers, is is still much less effort than it would have taken to write the *entire* analyzer by hand. Since we have concentrated on only those errors that are particularly disruptive to syntax analysis, the benefit is in fact larger than the gain in tagging accuracy reflects: although the category accuracy of the new tagger is only slightly better than in the original one, we were able to avoid half of the parsing errors caused by tagger errors.

4.2 Chunk parsing

4.2.1 Definition

The notion of *chunks* was originally proposed by Abney (1989). A chunk is defined as a sequence of a content word and its associated function words, such as "the bald man", "was sitting", or "on his suitcase". The central assertion is that human hearers process utterances as a *sequence* of chunks, where analysis of one chunk is largely independent of one another. Although some psychological evidence is cited for the existence of chunks (they are directly related to the assignment of stress in spoken language), their main motivation is to allow more efficient parsing.

According to Abney, the arrangement of words within chunks rigidly adheres to simple patterns that can easily be captured by context-free rules, while the attachment

[7] Tagging itself can be optimized to linear run-time by converting the completed tagger to a finite-state machine (Roche and Schabes, 1995).

of chunks to each other is considerably freer. For instance, lexical preference is a factor only in chunk attachment, and not in chunk delimitation. By distinguishing two stages in a parser, efficient techniques can be used for initial segmentation, while complications such as subcategorization are only necessary in the second stage. The scheme also reduces processing ambiguity in several ways. For instance, long noun phrases can be left unspecified with respect to their internal structure, since all possibilities are syntactically acceptable. Also, because of the two-stage strategy, the ambiguity of chunk delimitation is not multiplied with attachment ambiguity. Automatic parsers have since implemented the technique and found that it does contribute to more efficient parsing (Basili et al., 1998; Bartolini et al., 2004).

A chunk parsing component would seem very useful for a WCDG parser. Syntactic ambiguity 'on the small scale' contributes a considerable part of the ambiguity that so often threatens to overwhelm the selection process with its many possibilities. If sentences could be reliably partitioned into chunks with (say) three words each, and dependencies had to connect only chunks rather than individual words, this would reduce the number of dependency edges that have to be considered in the first place dramatically. For instance, a sentence with nine words would no longer allow $9 \cdot 8 = 72$ structurally different attachments between words, but only $3 \cdot 2 = 6$ different attachments between chunks. This is a reduction by over an order of magnitude, and since sentences are longer than nine words on the average, the computation would be even more favorable. Such a huge ambiguity reduction could allow the parser much more time to search for meaningful alternatives rather than keep rejecting word-to-word dependencies that should never have been allowed in the first place.

There is, of course, a catch to this hypothesis: Abney explicitly assumes that chunks cannot contain each other. For instance, the sentence "The bald man / was sitting / on his suitcase" contains three chunks, which cover the entire sentence but do not overlap. Language is assumed to be fundamentally not a *sequence* rather than a *structure* composed of chunks. This property is adduced to explain why English allows the construction "the man proud of his son", but not "the proud of his son man".

Unfortunately, this is not at all true of German. Consider the sentence

"Die Verfassung und **[das [von [den Organen] [der Union] [in Ausübung] [der [der Union] übertragenen Zuständigkeiten] gesetzte] Recht]** haben Vorrang vor dem Recht der Mitgliedstaaten."
(The Constitution and ***[the laws created] [by organs] [of the Union] [in the exercise] [of the authority vested] [in the Union]*** *take precedence over the laws of the member states.)*
(European Constitution, §I-6)

In the marked place, no fewer than three noun phrases are nested within each other; each of the lower noun phrases is licensed by a participial adjective which itself modifies the next noun. Note how the English gloss does separate these constructions

into a plain sequence of chunks. In German, however, all three are actually nested within each other, so that the first article modifies the last noun. Analysing this sentence in terms of chunks would force us either to include all words from 'das' to 'Recht' into a single chunk, which largely defeats the purpose of two-stage processing, or to assume chunks such as "das", which contain only a meaningless article. Of course, this could actually be done if one is merely interested in the potential algorithmic advantages, and not in the linguistic theory behind them. But reducing a sentence to very small chunks defeats the purpose of two-stage processing just as much as postulating very large chunks: allowing very long or even nested chunks makes the first stage (chunk delimitation) much more difficult, while trivial chunks such as "das" essentially transfer all work to the second stage.

This type of construction is not easy to understand even for native speakers, and is predominantly found in formal written language, but a broad-coverage analyzer cannot simply ignore it. Therefore, chunk boundaries cannot be used as hard filters in the same way as in English without the risk of reduced coverage. On the other hand, using chunk analysis as a guide rather than a filter does not reduce the problem size at all; it can only allow better navigation through the unchanged search space. Therefore it remains to be seen whether automatic chunk detection can be of any help for the German WCDG.

4.2.2 Integration into WCDG

While a part-of-speech tagger predicts only features about individual words, chunk parsing goes one step further and makes predictions about the relations between words. It does not explicitly suggest particular dependencies, but places restrictions on which dependencies can be established. How should these restrictions be expressed as WCDG constraints?

The original exposition (Abney, 1989) assumed that dependencies would be established between chunks rather than words. This cannot be implemented directly, since WCDG deals only with words and not with higher-order structures. Instead, it must be assumed that every chunk is represented by a *head word*, and that to subordinate a chunk means to subordinate its head word to a word outside the chunk. In contrast, all other words in the chunk must have their regents within the chunk. If we assume that the extent of each chunk is attached by the predictor under the keys 'start' and 'end' at each word, and the head word is marked by specifying the key 'head', the chunk principle can be cast into a single constraint:

```
{X!SYN} : chunker : 0.9 :
 X^from < predict(X@id, CP, start) |
 X^to   > predict(X@id, CP, end)
 <->
 predict(X@id, CP, head) = 1;
```

(In plain words: if a word is attached to a regent outside its own chunk, it must be a head word, and vice versa.)

The computation above assumed a stronger restriction: namely, that the regent of one head word must be another head word (in fact it assumed only head words can be modified at all, whether within or across chunks). However, this would contradict other modeling decisions already made in the grammar. For instance, in our model a complex auxiliary group such as "verkauft worden seien" (*had been sold*) forms a chain where the word 'verkauft' modifies the next word 'worden', rather than the head word 'seien'. Several other constructions also contradict the head attachment principle. However, in all of these cases the grammar already contains normalization constraints (although not always hard constraints) that enforce the opposite decision, so that a reduction of problem size is nevertheless achieved.

4.2.3 Experimental setup

For our experiments we employ the decision tree analyzer *TreeTagger* (Schmid, 1994b), which is freely available for research purposes. In addition to part of speech and lemma information, it is also capable of predicting chunk boundaries for running text. It contains a pre-computed model of German that inserts markers for nominal, verbal and prepositional chunks. However, it does not explicitly mark head words. Since a wrapper program is required anyway in order to transform the output of TreeTagger into the format that the WCDG predictor interface expects, we can assign the task of computing head words to this wrapper. Noun chunks are assumed to have the first word as their head, while verb phrases take the finite verb as head word. In prepositional phrases the first or last word is considered the head word, depending on whether a preposition or a postposition is present.

		Accuracy	
Length	Instances	structural	labelled
1 – 10	340	94.8%	92.8%
11 – 20	323	91.9%	90.0%
21 – 30	229	89.3%	87.8%
31 – 40	76	86.9%	85.2%
>40	32	83.4%	81.2%
overall	1,000	89.8%	88.0%

Table 4.9: Parsing results with the chunk constraint.

Table 4.9 shows the results of adding this constraint to the grammar and parsing the same sentences as before. Overall, the structural accuracy improves from 89.3% to 89.8%. Although this is a significant improvement, it is not a very big one: the attachment error rate decreases by only 5%.

There could be three different reasons why chunk predictions do not help our parser very much.

1. The 'bad idea' theory: Perhaps the predictor does not tell the parser anything that it does not already know. If WCDG tends to make errors in attaching head words to each other rather than in establishing the chunk-internal structure, then additional knowledge of chunk boundaries is of little use no matter how accurate it is, and we can never expect much help from a chunk predictor at all.

2. The 'bad integration' theory: The predictions might be integrated with the parser in a suboptimal way. Making the constraint too strong risks total failure of parsing if the input does contain errors; leaving it too lax could lose the guiding influence of the predictor entirely. There is no strong reason that the chosen constraint penalty of 0.9 should be the best one, except that it worked well in a much earlier version of the grammar.

3. The 'bad input' theory: the predictions themselves might be too unreliable to provide a benefit. A mispredicted chunk boundary can impede parsing just like a word misclassified by part-of-speech tagging, because it will punish at least one dependency edge that is required to construct the correct analysis. Indeed, in previous experiments it was found that errors in the chunk boundary assignments diminish the benefit of the component more than errors in part-of-speech tagging (Daum et al., 2003).

penalty	real chunks	ideal chunks
0.0	88.7%/86.9%	90.4%/88.6%
0.1	89.1%/87.2%	90.5%/88.7%
0.5	89.5%/87.7%	90.3%/88.5%
0.9	89.8%/88.0%	90.1%/88.4%

Table 4.10: Effects of chunk accuracy and integration strength.

The second and third theories can easily be tested: as with part-of-speech tagging, we can replace the TreeTagger with a simulated chunker that never makes mistakes at all, and observe the effect on parsing accuracy. Also, we can try different penalties for the integration constraint to see what effect they have. Table 4.10 shows the results of two series of experiments with different penalties, and with real chunks as obtained from TreeTagger as well as with 'ideal' chunks that are simply read off the annotation.

It can be seen that when using TreeTagger as a chunk predictor, the rather loose integration resulting from a constraint penalty of 0.9 is in fact the best value of those tested. Apparently, when more attention is paid to the predicted chunk boundaries,

the benefit of knowing the micro-structure of a sentence quickly turns into a disadvantage, as wrong predictions become harder and harder to overrule. Somewhere between a penalty of 0.5 (mild recommendation) and 0.1 (serious rule), parsing accuracy actually falls below the level measured without chunk parsing.

On the other hand, if we eliminate the influence of wrong predictions by simulating an ideal predictor, accuracy tends to rise with a tighter integration. However, it does not rise very far at all; the best measured figure of 90.5% structural accuracy is only slightly above the figure for perfect part-of-speech information shown in Table 3.2. This means that the first theory remains as the most fitting explanation: WCDG does not receive a large benefit from a chunk predictor because it does not make many errors below the chunk level as it is.

4.3 Supertagging

Part-of-speech tagging for natural language processing was a success story both in terms of solving the task itself and in leveraging it to aid full parsing. It was therefore natural to investigate whether the expressivity of such tags might profitably be extended to predict more than just syntactic categories.

4.3.1 Definition

The first application of this idea was reported by Joshi and Bangalore (1994), who used corpus statistics to predict the elementary trees of Lexicalized Tree-Adjoining Grammar (LTAG). The lexicon of this grammar formalism associates words not just with categories, but with tree fragments which must be combined by *substitution* and *adjunction* operations. This leads to a much higher lexical ambiguity than with ordinary part-of-speech tags, since the lexicon has to make distinctions not only between the noun and verb readings of an ambiguous word, but also between transitive and intransitive verb readings, between head and modifier nouns, etc. Statistical methods were used to resolve this ambiguity; up to 77% of elementary trees could be predicted correctly, and this resulted in a speedup of the parser proper of 87%. Prediction methods were later refined (Bangalore and Joshi, 1999) to the point where up to 92% of supertags could be predicted correctly.

The authors noted that supertagging in this domain amounts to 'almost parsing', since after determining a supertag sequence for a sentence, little ambiguity remains about the final tree structure. Nasr and Rambow (2004) quantify this claim by measuring the dependency accuracy of a parser that is supplied with perfect supertags, and find it to be 97.7% on the WSJ test set.

Another formalism that has profited from supertag predictions is *Combinatorial Categorial Grammar* (CCG). Clark and Curran (2004) modify a maximum entropy tagger to predict supertags suitable for their CCG parser, and conclude that it is only

through supertagging that their probability model can be trained at all on present-day hardware.

4.3.2 Supertags in WCDG

LTAG and CCG somewhat resemble each other in their general approach: they explicitly aim to provide linguistically adequate analyses of language, and for the purpose they use at least partly hand-written grammars that are more expressive than classical CFGs. They also define computational problems whose exact solution requires more complex algorithms than the polynomial-time methods usually employed. Finally, they profit from supertag information not just as a performance-enhancing component, but actually as an enabler that makes parsing practically feasible in the first place.

WCDG shares most of these features, and as we have seen, it also greatly profits from tagging techniques. In particular the XTAG system, as described by Bangalore et al. (1995), is conceptually rather similar to WCDG; for instance, their lightweight dependency analyzer uses global heuristics (such as 'prefer arguments over adjuncts' or 'prefer low to high PP attachment') for ranking alternative parses, which serve much the same function as WCDG's preference constraints, and in fact have direct equivalents in our grammar. It seems likely that WCDG could also profit from tagging at a higher level than mere syntax categories.

But in contrast to the previous examples, it is not immediately obvious how precisely to define a 'supertag' for a dependency parser, since WCDG does not operate on ready-made tree fragments, but on individual dependency edges only. Obviously, to have an impact beyond that already measured for part-of-speech tagging, words have to be classified into finer classes than syntax categories, but a trade-off is to be expected between expressiveness and predictability: a small number of classes may not be able to capture all trends that could be exploited by machine learning, but a large number of classes may lead to very sparse data that cannot be accurately learnt at all.

The minimal useful description of a tree fragment could be the syntactic label under which a word is subordinated; for words such as articles, not much benefit is to be expected from knowing that they will be labelled 'DET', since not many alternatives are available, but nouns and pronouns, for instance, could profit from being pre-tagged as subjects or objects (see p. 46). Another valuable piece of information would be to know the *direction* of a dependency (whether the word is subordinated as a pre- or post-modifier). For further distinction one could also tag each word with the labels of its dependent words. Again, pre-modifiers and post-modifiers could either be distinguished or treated uniformly. A supertag that describes the entire local environment of a word in this way would be roughly as informative as an elementary tree in LTAG.

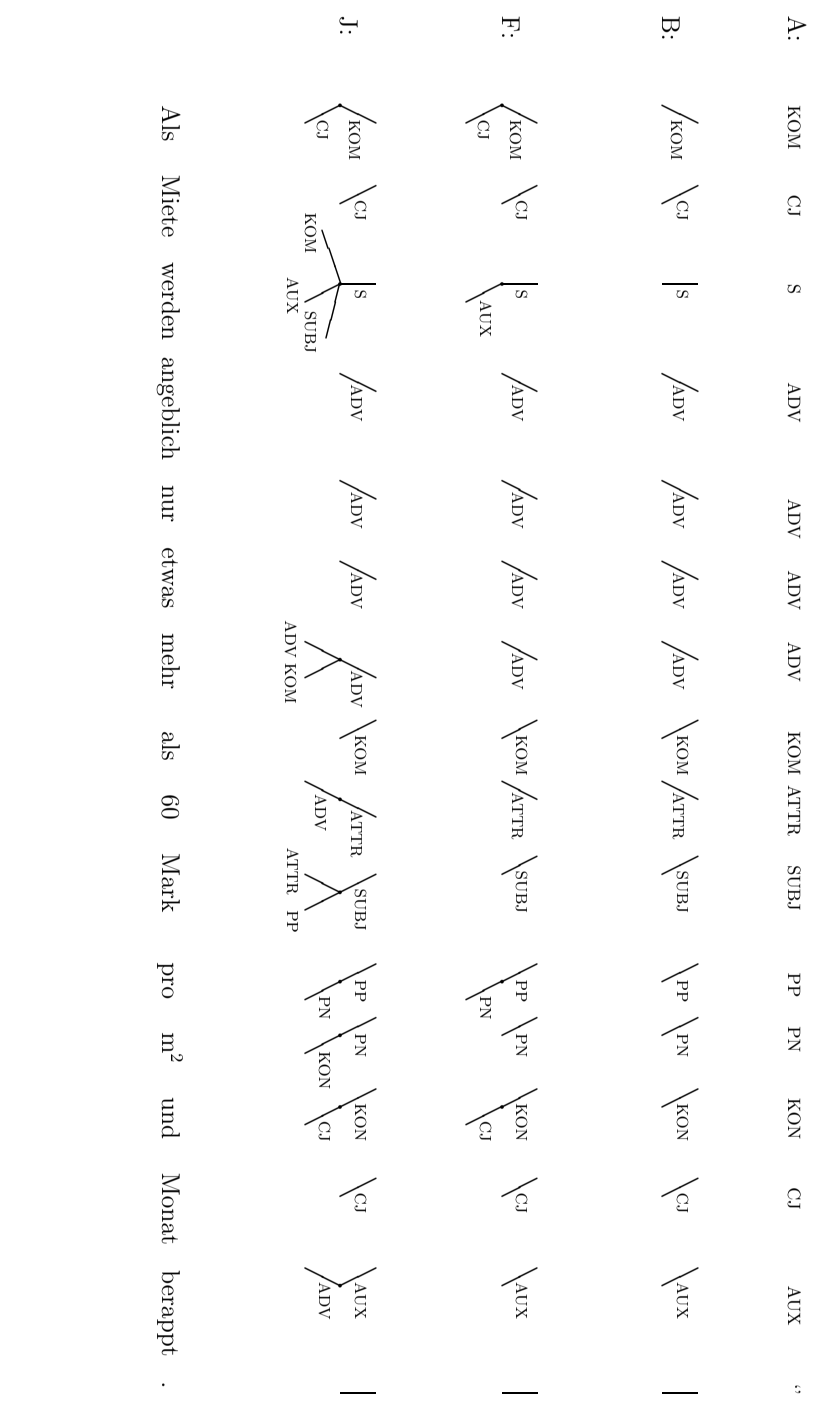

Figure 4.4: Correct supertags for the example sentence under different models.

An investigation into different variants of supertag definitions for WCDG was previously conducted by Foth et al. (2005b). Ten different supertag definitions were used which ranged from predicting just the syntax label of each word (model A) to predicting the label and direction of the word and all of its both adjuncts and complements (model J). As an example, consider the example sentence for POS tagging in Figure 4.3 (Section 4.1.6). Figure 4.4 shows the correct supertags that should be assigned to each word under the models A, B, F and J. Model A describes only the label of the dependency between each word and its regent, while model B describes the label and the direction of the dependency (represented here through a stylized tree fragment). Model F also describes the complements of each word, i.e. those words that depend on it in an obligatory relation. In the example, the complements of the comparator, the auxiliary verb, the preposition and the conjunction are depicted as dependencies below the primary dependency edge. Finally, Model J also describes the adjuncts of each word, and therefore *all* dependents are depicted.

Note that while in this figure the supertags resemble fragments of the dependency tree, they do not describe the local environment of each word quite as exhaustively as LTAG supertags would do. For instance, some models describe dependents, but do not distinguish between pre- and post-modifiers (C,D,G,H). Both the models F and J do make this distinction: for instance, the supertags of the word 'Als' express the fact that the dependent with the label `CJ` is a *post*-modifier. However, the relative order among several pre-modifiers is never described: the J-supertag of the word 'werden' describes two post-modifiers labelled `SUBJ` and `AUX`, but not the order in which they occur. Multiple dependents with the same label are also not explicitly represented: the word 'mehr' has two different pre-modifiers labelled `ADV`, but even the J-supertag does not express this.

describes: Model	edge label	edge direction	dependent labels	dependent directions	#tags	Supertag accuracy	Component accuracy
A	yes	no	none	no	35	84.1%	84.1%
B	yes	yes	none	no	73	78.9%	85.7%
C	yes	no	obligatory	no	914	81.1%	88.5%
D	yes	yes	obligatory	no	1336	76.9%	90.8%
E	yes	no	obligatory	yes	1465	80.6%	91.8%
F	yes	yes	obligatory	yes	2026	76.2%	90.9%
G	yes	no	all	no	6858	71.8%	81.3%
H	yes	yes	all	no	8684	67.9%	85.8%
I	yes	no	all	yes	10762	71.6%	84.3%
J	yes	yes	all	yes	12947	67.6%	84.5%

Table 4.11: Definition of different supertag models.

Foth et al. (2005b) retrained the program TnT (which was already used in Section 4.1) with supertags rather than categories as its input, and measured the precision of its predictions. The same test set as in this work was used, and the rest of the NEGRA and TIGER corpora was used as the training set in each case. Table 4.11 shows the

result of predicting supertags for each definition. Several different observations can be made:

- As the complexity of supertag definitions rises, the number of actually occurring tags also increases, but the training set becomes sparser. For model A supertags, which consist only of edge labels, evidence can be found for all 35 possible events; but model J supertags could theoretically occur in millions of varieties, of which less than 13,000 are observed.
- The accuracy of prediction tends to fall with the size of the tag set, from well above 80% to well below 70%.
- However, some predictions are much easier to make than others. Model B, which predicts only the label and subordination direction of each word, reaches a lower accuracy for the same training set than model C, which predicts the label of each word and the labels of its complements (but not adjuncts), even though the number of occurring supertags is much lower. (This is probably at least partly due to the semi-free word order of German.)

To abstract away from the general difficulty of predicting large tag sets, one can define a more fine-grained measure that distinguishes between completely correct, partially correct and completely wrong predictions. In our experiments, each supertag must make up to four independent partial predictions about a word w:

1. What label does w bear?
2. Is w a pre-modifier, a post-modifier, or a sentence root?
3. Which types of pre-modifiers occur under w?
4. Which types of post-modifiers occur under w?

We define the *component accuracy* of a predicted supertag as the number of its correct partial predictions divided by the number of all partial predictions that it makes. For model A, with only one component, the component accuracy of a single supertag is always 0 or 1, and is identical to the supertag accuracy; but for model B, with two components, a supertag can exhibit a component accuracy of 0.5. This reflects more closely the intuitive appraisal that a supertag which gets the label right and the direction wrong is at least half-correct, and better than a supertag that gets both wrong. For the more complicated models C, D, G, and H, the set of occurring modifier labels is also counted as one partial prediction, but the set must be entirely correct; a partially correct set is still treated as an incorrect partial prediction. For the models E, F, I, and J, pre-modifiers and post-modifiers are counted as two predictions, so if a model J supertag fails to predict one post-modifier,

its pre-modifier prediction can still be correct. The models C–J thus each have two to four components whose prediction is measured separately.

The component accuracy of an entire supertagger run can be straightforwardly defined as the average of the component accuracy of each predicted supertag. Under this measure, the intermediate models C–F achieve the highest accuracy; obviously, up to a point the richer information in the neighbouring supertags counteracts the general difficulty of predicting large tag sets.

Foth et al. (2005b) used the predictions made by the retrained TnT to add new constraints to WCDG and measured the effect on parsing accuracy. It was found that parsing benefits most from the most complex supertag variants, even though the precision of the supertags themselves is not the highest observed.

The previous work used supertag predictions to augment the grammar in a sentence-specific way for each parsing run; for instance, constraints such as 'the second word is labelled `DET`' or 'the fourth word has a pre-modifier labelled `OBJA`'. This allowed only a fixed set of sentences to be analysed. With a general predictor as developed in Section 4.1.4, the retrained supertagger can be applied to arbitrary input; its output is used by generic constraints that check the label, direction etc. of every edge in the tree in a uniform way.

Consider again the tree in Figure 4.3. Assuming the most complex supertag definition, and in the absence of prediction errors, the following predictions should be made about the word 'werden':

- Its syntactic label is `S`.
- It has no regent.
- It is modified by a `KOM` edge to its left.
- It is modified by a `SUBJ` and an `AUX` edge to its right (but not necessarily in this order).

The first two predictions can be straightforwardly translated into two isolated predictions, but the other two introduce a complication: the set of possible pre- and post-modifier combinations of a given word would be impractically large if it were treated as a single prediction, since even disregarding order there is an exponential number of combinations. It is much easier to treat the occurrence of each modifier as a separate prediction which can be checked individually. On the other hand, this would theoretically require 70 individual predictions to be made and 70 new constraints to be checked for each word (twice the size of our label set). As a compromise, we make only a fixed number n of predictions on either side and ignore the rare cases when a word has pre- or post-modifiers with more than n different labels.

In the example, the supertag for the word 'werden' would be translated into the following predictions for $n = 4$:

- label: S
- direction: N
- pre1: KOM
- pre2: none
- pre3: none
- pre4: none
- post1: SUBJ
- post2: AUX
- post3: none
- post4: none

The prediction 'none' is to be interpreted as 'fewer than i different labels were predicted on this side'. These predictions can now be trivially checked by general constraints; as an example, the check of the predictions 'label' and 'pre1' can be performed in the following way:

```
{X!SYN} : 'ST:label' :
  predict(X@id, ST, 'label') = X.label;

{X:SYN} : 'ST:missing left dep' :
  predict(X@id, ST, pre1) != none
  ->
  has(X@id, predict(X@id, ST, pre1));
```

A recurring question is of course how much weight to give these constraints. Since Foth et al. (2005b) obtained the best results with a constraint penalty of 0.9, we reuse this value in the following experiments. For comparison with other work, we also perform a control experiment in which perfect supertags are assumed; to measure the maximal benefit that supertagging can theoretically provide, we then re-run this experiment with a penalty of 0, and finally we observe the effect of this penalty when using real supertags again. Since a very few of the sentences in our test set do actually contain words with more than four different modifications on one side, we increase the value of n to 6 in the experiments with perfect supertags, in order to avoid forbidding the correct structure with a hard constraint; the real supertags are still trained and tested with a value of $n = 4$.

The most important of the results in Table 4.12 is certainly that of experiment 2: even automatically computed supertags are very valuable to WCDG parsing accuracy, in

Experiment	Supertags	Penalty	Accuracy (on covered sentences)	Coverage
1	none	-	89.3%/87.5%	100%
2	real	0.9	91.9%/90.5%	100%
3	ideal	0.9	97.2%/96.7%	100%
4	ideal	0.0	98.5%/98.4%	99%
5	real	0.0	91.4%/90.3%	16%

Table 4.12: Effects of supertagging on parsing the test set.

fact they lead to the greatest improvement over the baseline grammar measured yet. This is despite the comparatively low accuracy of the supertags themselves; recall that one out of three of the complex supertags used here is wrong in some way, and even when measuring the more pertinent component accuracy, one out of six predictions is still wrong. Nevertheless, their cumulative effect is enough to raise the syntactic attachment accuracy far above 90%. Clearly, the distributional evidence contributed by supertags provides information that the general grammar rules do not capture as well.

The effects of supertags have often been investigated under the assumption that the correct supertags are available to the parser; the consistent result has been that perfect supertagging would *almost* eliminate the need for full parsing (Bangalore and Joshi, 1999; Nasr and Rambow, 2004). Experiment 3 confirms this assessment; when the predictions extracted from supertags are consistently correct, the attachment error rate drops to 2.8%. It can be reduced to a mere 1.5% by making the supertag constraints hard, so that they overrule most other rules. The remaining errors are mainly due to two phenomena: first, even our complex supertag model does not entirely determine syntax structure; for instance, if two nearby words both carry several adverbial pre-modifiers, then at least one of the adverbs can be misattached and still satisfy all predictions. This could be avoided only by an even finer supertag set that predicts both the order and number of each type of modifier for each word. Second, hard supertag constraints can sometimes conflict with hard part-of-speech tagging predictions; if the part-of-speech tagger makes an error that causes the predicted edge label to become impossible, then the priority between the two hard constraints is not well-defined, and the parsing problem can even become unsolvable. Note that only 99% of all sentences could be analysed in experiment 4.

Making the relevant constraints hard corresponds to the more usual use of supertags as a *filtering* rather than a *guiding* component: usually, the elementary trees predicted by the supertaggers are the *only* ones that the parser may use. This means that discarding a necessary supertag altogether prevents the parser from ever finding the correct solution. As a countermeasure, more than one supertag is often fed into a parser, for instance the n most probable ones according to the statistical model (Bangalore and Joshi, 1999), or even all those which cannot be proved to be irrelevant (Boullier, 2003). Our experiment 5 simulates this mode of operation, but not the

countermeasure: the *real* supertagger is integrated with hard constraints, but with the minimal value of $n = 1$. Although this still results in an accuracy improvement over the baseline experiment, far too many parsing problems become impossible to solve; the coverage drops to only 16%, which is quite inadequate for our goals. A larger value for n would probably be more useful; however, this would require a much more complicated translation process from a set of supertags to an ensemble of alternative predictions, and also more complex constraints to check them. As with other sources of statistical knowledge, the better solution seems to lie in not giving the predictions too much weight in the first place.

4.4 PP attachment

4.4.1 Structural predictions

All oracles discussed so far were variants of *classifiers* which sort words in the input into one of several classes: part-of-speech tagging sorts words into syntactic categories; chunk parsing sorts them into chunk heads, chunk-start words, chunk-stop words and chunk-internal words; and even supertagging ultimately only sorts words into much finer classes. Each of these classifications proved useful as an aid to parsing, but none actually predicted the word-to-word subordinations themselves, which parsing is supposed to construct.

Often the choice is obvious, given the prediction. A supertag specifying 'this word is modified by a preceding determiner' is usually selected because an article immediately precedes, and the recommendation can only be followed by in fact subordinating that article to the word. But there are also subordinations that still allow a considerable amount of ambiguity even if all predictions are followed. Relations such as adverbial modifiers, relative clauses, or co-ordinations typically have more than one possible attachment point in a sentence; therefore predictions about the precise regent of individual words could be valuable.

For instance, the hard definition constraints about modifiers are predominantly category-based (for instance, adverbial modifiers can modify verbs, nouns and adjectives), which usually means that more than one regent remains possible. Our model of German as it is attempts to define a ranking even between such ambiguities, but often it cannot make informed choices and has to rely on general principles that have many exceptions. For instance, a very weak rule is to prefer a closer regent over a distant one. This principle is in fact implemented as a preference constraint, and it is a useful one because overall, it tends to disambiguate correctly more often than not, but there are still very many cases where it predicts the wrong regent.

It seems likely that rules of an intermediate nature — specific predictions about word-to-word relations that are not absolute, but more reliable than our very vague preferences — could help analysing such modifiers correctly. This would require a

closer distinction between words than that based on their category; in other words, it would require *lexicalized* rules.

4.4.2 Prepositions as an error source

As a promising candidate for modifiers that could profit from lexicalized preferences, we now develop a component that helps disambiguate the regents of prepositions. Not without reason has this task been a poster child of syntactical ambiguity. The array of statistical methods that have been applied to solving it is almost as varied as for part-of-speech tagging: it includes handwritten deterministic rules (Whittemore et al., 1990), maximum-likelihood estimates from supervised learning (Hindle and Rooth, 1991), maximum-entropy models (Ratnaparkhi and Roukos, 1994), transformation-based learning (Brill and Resnik, 1994), statistical decision trees (Stetina and Nagao, 1997), and neural networks (Sopena et al., 1998). Various knowledge sources have been used to obtain the necessary estimates, such as as annotated text (Collins and Brooks, 1995), unannotated text (Ratnaparkhi, 1998), web search engine queries (Pavia and Aït-Mokhtar, 2003), parallel corpora (Schwartz et al., 2003), or machine-readable thesauruses (McLauchlan, 2004). Even the best efforts so far achieve less than 90% of accuracy on unseen input.

Label	occurred	retrieved	percentage	no. of errors
	2350	2350	100.0	0
DET	2030	2001	98.6	29
PN	1725	1684	97.6	41
PP	1695	1133	66.8	562
ADV	1235	936	75.8	299
SUBJ	1210	1130	93.4	80
ATTR	1143	1106	96.8	36
S	1142	997	87.3	145
AUX	635	595	93.7	40
OBJA	604	522	86.4	82

Table 4.13: Attachment accuracy by annotated label on the test set.

The amount of work that has been expended on the task indicates how difficult it is; we can confirm this difficulty for our parser by measuring how well it does on particular types of relations. Table 4.13 measures the structural accuracy of the experiment from Table 3.2 conditioned on the *annotated* label of each word. For instance, the most common type of edge, annotated with the empty string as a label, occurs only with punctuation; since there is virtually no ambiguity about which words are or are not punctuation, all 2350 cases were correctly left unattached. Similarly, the label 'DET' labels the dependency between a determiner and its noun, which is very easy to retrieve: of the 2030 words labelled 'DET' in the test set, the parser attached 2001 correctly.

In contrast, the label 'PP' exhibits the lowest structural recall of all important dependency types. Note that this figure applies in the context of parsing entire sequences and cannot directly be compared with most previous work on prepositional attachment, where only two possible regents are allowed in the first place, of which one is always correct. The WCDG parser has to select the correct regent for a preposition from *all* words in each sentence. This kind of situated PP attachment accounts for the largest number of attachment errors in absolute terms: roughly one word in 30 is a preposition that the parser attaches incorrectly. This means that information about PP attachment could add up to 3% to the overall structural accuracy. The next most serious problem is the similar adverb attachment, which contributes about half as many errors.

4.4.3 Possible disambiguation criteria

The principal reason for the comparatively poor performance on PP subordination is of course the wide range of possibilities that the individual preposition faces in many utterances. Modern German exhibits some features that particularly contribute to the difficulty. For instance, German prepositions can attach to adjective, noun and verb phrases, both to the left and to the right. Also, the separable verbs of German add a particular complication: the presence or absence of a particular prefix can influence the meaning and hence the behaviour of a full verb considerably, so that a verb attachment cannot adequately be judged by considering only one dependency relation at a time.

Nevertheless, speakers of German are probably not more confused about the relations of prepositions in a sentence than speakers of other languages. A variety of factors from different levels of understanding contributes to their ability to disambiguate PP attachment successfully.

- Syntax:

  "SEGA hat **für den 14. Oktober** den Verkaufsstart der internetfähigen Spielekonsole Dreamcast angekündigt."
  (SEGA has announced that sale of its internet-enabled console Dreamcast will begin on October 14th.)
  (heiseticker, item 6018)

  Although prepositions can attach both to preceding and to following words, and to both nouns and verbs, attachment to a following noun is unusual. Therefore the marked prepositional phrase is more likely to modify the verb.

- Topological fields:

  "Die Version **für Windows NT** soll im November folgen."
  (The Windows NT version will follow in November.)
  (heiseticker, item 6486)

Both the subject 'Version' and the preposition 'für' could plausibly modify the verb 'soll', but not both: since only one constituent usually precedes the finite verb, the preposition must be assumed to modify the subject rather than the verb.

- Default reasoning:

  "Ich behalte die Karte **für morgen**."
  (I keep the ticket for tomorrow.)
  (constructed)

  The speaker might be talking about a dedicated ticket for tomorrow's event, or about a generic ticket that may or may not get used tomorrow. Although both interpretations are plausible, the latter one appears to be preferred. This accords with the overall preference for verb attachment that treebank statistics predict.

- Idiomatic expressions:

  "Die Kommission achtet **auf die Anwendung dieses Artikels** und erlässt erforderlichenfalls geeignete Europäische Verordnungen oder Beschlüsse."
  (The Commission attends to the application of this article and issues appropriate European ordinances or resolutions if necessary.)
  (European Constitution, §III-166)

  Since the verb 'achten' can form an idiomatic group with the preposition 'auf', it is likely that a co-occurrence of both actually is an instance of that idiom, because non-idiomatic homonyms are generally less frequent than their idiomatic counterparts (Nunberg et al., 1994).

- General lexical preference:

  "Für das Paket zahlte die Telekom damals einen Betrag **von drei Milliarden Mark**."
  (Telekom had paid DM 3,000,000,000 for the bundle.)
  (heiseticker, item 19998)

  Here the preposition 'von' has no obvious role to play when combined with the verb 'zahlen', because payment does not involve an obvious 'topic' slot that a monetary amount could fill. However, it *is* often used to specify the amount of the sum of money itself. Therefore, noun attachment is more likely in this case despite the global preference for verb attachment.

  We see that the preference of particular lexical items to combine with each other is not at all limited to idiomatic expressions, but can also disambiguate between two free combinations that are equally possible but not equally salient.

- Named entities:

"Davon sind 12 Prozent bei der Kreditanstalt **für Wiederaufbau** geparkt."
(12% of these reside with the Kreditanstalt für Wiederaufbau.*)*
(heiseticker, item 18197)

This utterance will invariably be understood as a noun attachment by anyone who is familiar with the governmental agency 'Kreditanstalt für Wiederaufbau'.

- Style:

  "Außerdem verpflichtete sich das Unternehmen, insgesamt rund 8 Millionen weitere Aktien von Lycos Europe **für 10 Euro pro Aktie** zu zeichnen."
  (The company also pledged to tender 8,000,000 additional shares of Lycos Europe for e*10 a share.)*
  (heiseticker, item 13982)

  Here, the combination of 'Aktie and 'für' is highly salient, but the noun attachment would create a noun phrase with a repeated element ('shares for 10 e per share'), which is considered bad style. If this reading were intended, the formulation should have been "für jeweils 10 Euro" ('for 10 e each').

  "AT&T verlangt daher von der FCC, dass diese Behörde der Fusion von Time Warner und AOL nur zustimmt, wenn Time Warner einwilligt, die 25,5 Prozent **von TWE von AT&T** zurückzukaufen."
  (Therefore, AT&T demands that the FCC should allow the merger between Time Warner and AOL only if Time Warner agrees to buy back the 25.5% of shares in TWE from AOL.)
  (heiseticker, item 13059)

  Both marked prepositional phrases could fittingly modify the preceding noun 'Prozent' or the following verb 'zurückzukaufen', but they should not modify the same word; if there really were two different sellers, good style requires the use of explicit co-ordination (or at least a comma) rather than two parallel prepositional qualifications. Therefore the usual interpretation would be that the first preposition subordinates under the preceding noun and the second, under the following verb (the inverse distribution is prohibited by the projectivity principle).

- Semantic weakness:

  "Das viel zitierte "Ende des PC" kommt nach Meinung **von Barrett** jedenfalls nicht so bald."
  (According to Barrett, the "end of the PC", often predicted, will not arrive in the near future.)
  (heiseticker, item 18674)

  Here, the expression 'nach Meinung' ('following the opinion') is a strong indicator of noun attachment of the *following* preposition: its compositional meaning has faded, and it functions merely as a synonym for 'according to'. This meaning requires an argument, which is supplied by the preposition 'von'; the use of

'nach Meinung' without either a genitive or prepositional subordination would be unacceptable to almost the same degree as the use of a preposition without a kernel. Note that both the preposition and the noun behave quite normally with other words; the effect is characteristic only of the entire phrase and not of either of its components.

- World knowledge:

  "Bereits in wenigen Monaten steht nämlich der Übergang zu einer im 0,13-µm-Prozess gefertigten Pentium-4-Version (Codename "Northwood") an, die nur in Mainboards **mit dem neuen µPGA-478-Sockel** arbeiten wird."
  (The transition to a version of the Pentium 4 produced with the 0.13µm process (code-named 'Northwood'), which will only work in main boards with the new µPGA 478 socket, is already scheduled in a few months' time.)
  (heiseticker, item 17254)

  Here, the attachment of 'mit' to 'Mainboards' leads to the interpretation of the sentence given in the gloss. Attaching it to 'arbeiten' would mean that the new processor version will only co-operate with the new socket if it is used within a main board (implying that it might be useful even without a main board, just not in tandem with the new socket). This is obviously not the intended meaning, since computer processors cannot operate at all unless inserted in a main board, and in that case they do not co-operate with a socket, but reside within it. However, a very detailed representation of technical knowledge and advanced reasoning capabilities would be necessary for a parser that could duplicate this argument.

This list of factors in determining prepositional phrase attachment, by no means complete, conveys an impression of the manifold criteria that a perfect language analysis system would have to employ. The actual state of the art is, of course, more limited: so far, our model of German employs only the first five of these criteria for disambiguation.

Syntactical aspects can be dealt with most easily; for instance, it suffices to write a rule that constrains preposition-to-noun subordination to instances where the preposition follows the noun.[8] Also, in the context of a particular partial syntax tree, many more attachments are ruled out by the general projectivity and topology constraints. Similarly, default reasoning can be incorporated into a rule set by simply writing a weak constraint that is conditioned merely on the category of the attachment point.

A far greater effort has to be made to incorporate lexical preferences. The information that a particular prepositional attachment has idiomatic status depends crucially

[8]As usual, the prohibition is not absolute. For one, the ordering requirement might be violated in spontaneous speech with little loss in acceptability, and a grammar might want to capture this flexibility. Also, prepositions are allowed to precede nouns if both are part of an argument cluster within an elliptical co-ordination. Nevertheless these conditions are comparatively easy to describe compared with the vaguer criteria given above.

on the identity of both the preposition and the regent involved (and to a lesser degree upon the identity of the kernel noun). In the compilation of our verb lexicon, valence information was added manually to each of the about 8,000 German verbs covered there. About 200 of them were judged to form clear idiomatic expressions together with specific prepositions. But lexicalized rules are not only necessary to describe idiomatic associations. It is rather plausible intuitively that one of the two different situations that correspond to such an attachment ambiguity would be inherently more likely, as in the last example above. Furthermore, simplified versions of the attachment problem have often been solved solely on the basis of the main content words involved, i.e. disregarding the complete sentence (Hindle and Rooth, 1991; Ratnaparkhi and Roukos, 1994; Brill and Resnik, 1994; Collins and Brooks, 1995). It turns out that over 80% of these simplified problems can be solved correctly. This is a strong confirmation of the hypothesis that the words directly involved in a PP attachment are by far the most important factors to be considered.

4.4.4 Machine Learning for PP attachment

When the lexicon of German was compiled, a strict separation was maintained between adjunct and complement prepositions, reflected in different syntactic edge labels (PP versus OBJP). The identification of prepositional complements is a difficult task and subject to much individual disagreement; for instance, Albert et al. (2003) give six pages of possible criteria to distinguish the two, and ultimately refer to an extensive list of confirmed prepositional objects as the final test. The available lexicon takes a cautious approach and only considers very clear cases of idiomatic collocation to be prepositional objects; the great majority of all prepositional attachments are considered normal modifiers (PP). Therefore, even though OBJP edges are slightly preferred by the grammar, so far this criterion has little influence on overall disambiguation.

For instance, the compound verb ‘anfangen’ (to start) can take a direct object that denotes what is started. Alternatively, it can take the preposition ‘mit’ to express the same thing, but only if no direct object is present. This is a clear indicator of a non-compositional meaning, and therefore the lexicon specifies that ‘anfangen’ can take a prepositional object if the preposition is ‘mit’. Currently the only effect that this has on parsing is a slight preference for the attachment of ‘mit’ to ‘anfangen’, if both occur in the same sentence. This arises because complements are punished less than modifiers by the general distance constraints, and therefore an OBJP subordination will usually carry a better score than any PP subordination in the same sentence.

All other German prepositions are considered interchangeable with respect to this verb, which is surely an oversimplification. However, because of the sheer number of different verbs and prepositions a detailed description of general verb/preposition preference would be much more time-consuming than the restricted set of object relations. It would also be much more prone to individual error and revision, since all

the easy cases are already done. Clearly, these circumstances provide a good opportunity for machine learning: if enough data can be exploited, the association between prepositions and content words could be taken into account by a variable-strength constraint which is instantiated by values estimated from previous observations.

Throughout the following discussion we will use the following (constructed) example:

"Die Firmen müssen noch die Bedenken der EU-Kommission gegen die Fusion ausräumen."
(The companies have yet to address the Commission's concerns about the merger.)
(constructed)

The preposition 'gegen' here has three superficially plausible attachment points ('Bedenken', 'EU-Kommission', or 'ausräumen'), but the intended reading is almost certainly 'Bedenken gegen die Fusion' *(concerns about the merger)*. Note how this contradicts the more general heuristic decision rules: the attachment is to a noun rather than to the verb, and in fact to the more distant of the two nouns. Nevertheless, most informants would unhesitatingly choose the longer subordination, presumably because of world knowledge, which we cannot easily represent. However, it certainly seems as if a lexical preference exists between the two words 'Bedenken' and 'gegen' that is quite independent of the particulars of European economic policy, and that we ought to be able to measure in other German texts.

The obvious method of measuring the attraction of a word for another is to count how often each of them occurs, and how often the two form a dependency relation. When we examine all manually annotated dependency trees of German at our disposal, we do in fact find that neither 'ausräumen' nor 'Kommission' are ever modified by the preposition 'gegen', but 'Bedenken' is modified by it eight times. This seems to support our intuitive judgement, but care has to be taken to verify that the argument is valid. For instance, the collocation 'Bedenken'+'gegen' might occur more often than 'ausräumen'+'gegen' simply because the word 'Bedenken' is more common than 'ausräumen'. It turns out that the word 'Bedenken' on its own is, in fact, more frequent than 'ausräumen', but it is also much less frequent than the word 'Kommission'; this strengthens the case for the inferred association between this specific word and the preposition. To formalize the argument, an association between a word w and a preposition p should be considered stronger the more frequent it is (f_{w+p}), but weaker the more frequent its component words are in the first place (f_w, f_p). We can normalize the obtained value by division with the number of all words t, and obtain a measure of *lexical association strength*:

$$LA(w,p) := \frac{f_{w+p}}{\frac{f_w}{t} \cdot \frac{f_p}{t}}$$

The denominator of this expression is the *expected value* of co-occurrence between two words, that is, the number of times we would expect them to co-occur if there were no lexical preferences at all, while the numerator is the actually observed value.

If the ratio is much higher than 1, we can assume that the words are in fact positively correlated, while a number close to 0 indicates that the two words are particularly unlikely to form a subordination.

In our example, the desired subordination 'Bedenken'+'gegen' achieves a strength of 4.97, while both alternatives score 0. This measure, then, provides us with an *objective* reason for preferring the desired dependency in the example sentence, one that can be implemented in a parser without asking a native speaker for advice. We can add this sort of preference to our grammar by writing a new constraint with a penalty that varies with the measure of lexical association strength:

```
{X!SYN} : 'PP-attachment' : stat : [ predict(X@id, PP, X^from) ] :
  X.label = PP | X.label = KOM
  ->
  predict(X@id, PP, X^from) = 1;
```

This assumes that the result of these frequency calculations is made available to WCDG through a predictor program 'PP', which compares the evidence for all possible subordinations and makes corresponding predictions, scaled to the interval [0..1], that can be used as constraint penalties.

So far we have only considered *supervised* learning: heuristics about creating subordinations were derived from observing previous attachments. However, the amount of labelled training data available for this task is comparatively small. Annotating syntax structure is a rather expensive task, and is feasible only in limited amounts. The 100,000 syntax trees we have available are simply not enough to measure many common collocations: although we can make a case for a near-idiomatic combination such as 'Bedenken'+'gegen', the similar 'Bedenken'+'wegen' is not attested in our data at all, and therefore cannot be measured. Even the evidence for the example pair is rather marginal; eight occurrences could well be a chance result rather than the strong association we look for, particularly since all eight stem from the same area of the 'heiseticker' corpus (reporting about intended mergers during the 2000 dot.com bubble). If this section of our corpus had ended 20% earlier, we would not have a single instance to go on.

It is therefore inevitable that we should also consider *unsupervised* learning. This adds another level of uncertainty to the method: we can only count *co-occurrences* rather than attested *relations*, which may or may not indicate a lexical association. However, since WCDG is perfectly capable of profiting from uncertain knowledge, it will certainly be preferable to no knowledge at all. The upside of using raw text is that it can be obtained in far larger quantities, so that many more instances can be found.

For our experiments, we use the machine-readable corpus of back issues of the German daily newspaper *tageszeitung*, which can be licensed for research purposes. This

contains about 18,000,000 sentences with 295,000,000 words, which is more than two orders of magnitude more text than our dependency corpus; for instance, it contains the word 'Bedenken' 9,618 times, rather than just 37 times.

An important question is now which of the co-occurrences in raw text we should assume to be actual subordinations, and which must be considered accidental. A valuable indicator could be the order of words; for instance, dependency relations generally tend to occur between near rather than remote co-occurrences, and prepositions usually attach only two preceding rather than to following nouns. Following Volk (2001), we filter the set of possible subordinations as follows:

1. Prepositions are considered possible verb modifiers if they occur within 11 words on either side.
2. Prepositions are only considered possible noun modifiers if they occur immediately after the noun.
3. To counter the bias against nouns that step 2 introduces, the counts of prepositions immediately after nouns are artificially inflated by a factor of 5.

To reduce the parameter space as much as possible, we reduce each word to its lemma before counting it, so that 'ausräumen', 'ausgeräumt' and even the separated form 'räumt... aus' are counted as instances of the same verb[9]. Of course, in raw text this information is not as certain as in our annotated sentences; for instance 'ziehen' could be the infinitive of the full verb 'ziehen', or the (very rare) imperfect finite form of the verb 'zeihen'. Where we detect such possible conflicts, we inhibit the stemming algorithm so that counts of the two verbs will not be mixed up. For the same reason, we use the WCDG lexicon to undo noun compounding: both marked and unmarked compounds of known nouns are reduced to the base noun, on the assumption that lexical association is influenced much more by the base noun than by prefixes. Both 'EU-Kommission' and 'Untersuchungskommission' can therefore profit from evidence obtained about the base noun 'Kommission'.

As expected, this larger corpus contains many more instances of the pairs in question. The lexical association strength for the intended 'Bedenken'+'gegen' is in fact 24.79, that is, these words co-occur much more often than they should if there were no lexical preferences. The strength of association with the word 'Kommission' is only 0.66, and that of the word 'ausräumen' is 1.66. These figures are encouraging: even though we can only observe co-occurrence of words, rather than confirmed subordinations, the data confirm our intuitive judgement, in fact even stronger than in the supervised case.

[9]Preliminary experiments showed that reassembling separable verbs can be performed with 99% accuracy by simply associating each separated prefix with the nearest preceding full verb.

4.4.5 Experiment

In order to turn corpus counts into an actual predictor for the sake of WCDG, several implementation details have to be determined. It is probable that each of these could have a measurable impact on the accuracy of the predictions in themselves; nevertheless we do not attempt to optimize all parameters of the predictor, since this would require many additional runs across the entire training corpus. Instead we aim only for choices that are good enough to result in a positive benefit to syntax analysis, with the understanding that an improved predictor might yield better results.

First of all, should we base our decisions on unsupervised or supervised counts? We have seen that unsupervised counts can make predictions for more of the word combinations that might occur, but on the other hand supervised counts guarantee that we really are counting syntactic dependencies, and not co-occurrences that might be caused by misleading indirect correlation. In order to exploit both knowledge sources, we test possible preposition-word pairs first against the supervised counts. If the hypothetical regent word has occurred fewer than 1,000 times altogether, we switch instead to unsupervised counts. (This threshold value is the first of the several parameters that could probably be optimized for some additional benefit.)

The lexical association strength as defined in Section 4.4.4 is then computed for each possible regent of each preposition. A problem occurs when we have no evidence at all for a particular combination: should we assume that the words are in fact infinitely unlikely to modify each other, or that there is in fact some affinity that our data were too sparse to measure? Once again we attempt a compromise: if the expected number of co-occurrences is below 4, we assume that we have indeed measured a total lack of attraction; otherwise we *back off* from the identity of the regent to its part-of-speech tag, which always furnishes enough counts to have a rough idea of its behaviour.

In order to be used as a constraint penalty, the LA scores must then be transferred into the narrower range between 0 and 1. Since lexical attraction is, at best, a weak preference that can be preempted by many other language rules, and we expect a decisive influence only when no other important rules apply, it seems advisable to choose penalties generally close to 1. We choose the following mapping:

$$p(w,p) = \frac{\max(1, \min(0.8, 1 - (2 - \log_3(LA(w,p)))/50))}{c}$$

The lexical association strength contributes only logarithmically to the attachment preference. Overall, the possible penalties range from 0.8 to 1.0. The normalization constant c is chosen so that the highest of all values is exactly 1, as with part-of-speech tagging; this ensures that the preferred interpretation of a PP attachment incurs no penalty at all.

Some individual peculiarities of German must also be taken into account. For instance, the words ‘durch’ and ‘von’ are syntactically prepositions, but they also

function as passive morphemes (German has no synthetic passive voice), and then modify all past participles indifferently; therefore we exempt these subordinations altogether from penalty calculations.

Besides adding the new constraint 'PP attachment' to the grammar, we also disable several of the existing constraints that apply to prepositions, on the assumption that our lexicalized model is more exact than the very general preferences made so far. For instance, the global preference for verb attachment to noun attachment was essentially a primitive unlexicalized approximation of the new constraint, whose task is now taken over entirely by the statistical predictor. We also assume that lexical preference exerts a stronger influence on attachment than mere linear distance when both can apply to a configuration; therefore we change the distance constraint so that it exempts prepositions from the normal distance penalties imposed on adjuncts.

Class	without	with	change
1–10	94.4%	94.9%	+0.6%
11–20	91.1%	93.0%	+2.0%
21–30	89.0%	90.5%	+1.7%
31–40	87.0%	86.7%	−0.4%
>40	81.9%	84.7%	+3.3%
all	89.3%	90.6%	+1.5%

Table 4.14: Syntactical attachment accuracy without and with PP predictions.

Table 4.14 compares the results of parsing the test set with and without the new predictor. The most important result is that accuracy increases in almost all classes (only the second longest class of sentences, with only 76 instances, shows a slight decrease), and the overall attachment accuracy improves by 1.5%. This is a considerable improvement from our simple predictor, even if it is somewhat less than half of the 3% that we computed as the theoretical maximum benefit we could have achieved.

A brief closer look at the effects of the new constraint is therefore in order. The very first sentence of the test set furnishes both positive and negative examples:

"In Rosbach kann die Stadt das Baugrundstück im Wert von 870000 kostenfrei zur Verfügung stellen, außerdem erhält der Bauträger ein Darlehen von 579000 Mark."
(The city of Rosbach can supply the plot worth [DM] 870,000 free of charge; the contractor is also loaned DM 579,000.)
(NEGRA, sentence 18602)

The analysis without PP predictions contained two errors, both of them concerned with prepositions: the words 'zur' and 'von' were misattached. Both errors are corrected by the additional information because the collocations 'Darlehen'+'von' and 'zu'+'stellen' are rather strong, in fact almost idiomatic, and unsupervised learning has retrieved strong preferences for them. On the other hand, the augmented grammar also makes a new error: the word 'im' is attached to 'stellen' rather than the

correct 'Baugrundstück', since the verb 'stellen' (*put*) is strongly attracted to the local preposition 'in'.

This example alone points out at least two different weaknesses that hamper our simple model of PP attachment. Most importantly, it assumes that only the preposition itself and not its kernel NP influences the attachment preference. This is manifestly false: the complete phrase 'im Wert von', which simply means *worth* can be expected to exhibit a much stronger, in fact almost idiomatic, attraction for the noun 'Baugrundstück' than for the verb 'stellen'; this effect would reverse the error if it could be learnt automatically. We have disregarded such context effects only because for our data sets, counting entire prepositional phrases would increase the parameter space to the point where all counts can no longer easily be kept in RAM, and the extraction tools would have to be thoroughly rewritten. For optimal performance a PP attacher should certainly condition at least on the kernel noun of every preposition, and in fact most specialized work on PP attachment does exactly that.

A second weakness that contributes to this error is that we assume all prepositions with the same base form to be equivalent. Again this is generally true, but obviously untrue in this case. The form 'im' is used only as the dative variety of 'in' with masculine or neuter nouns, and therefore only if the preposition is used in the *essive* (local) sense, while the strong attraction of 'stellen' for 'in' stems mainly from the *lative* (directional) sense. Again, distinguishing between both varieties would probably have lessened the misleading attraction and reversed the error. Unfortunately, the preposition 'in' is atypical in this regard with its special forms 'im' for essive and 'ins' for lative use. Most other prepositions do not allow such deductions to be made easily in raw text because they have only one form, and therefore our counts ignored the distinction.

Yet another systematic weakness occurs only for the unsupervised predictions. We have seen that in the initial example sentence, unsupervised learning correctly preferred the desired pair 'Bedenken'+'gegen', but in fact it also computed a LA above 1 for the wrong pair 'ausräumen'+'gegen', although no common constructions spring to mind where these two words are indeed subordinated under each other. When we check the evidence for this figure, we find that it exclusively derives from cases in which 'ausräumen' and 'gegen' co-occur, but do not form a dependency relation; that is, the score obtained for this pair was based *entirely* on failed heuristics.

Many of these cases are sentences which combine all three words: the verb 'ausräumen' has a high lexical preference for the noun 'Bedenken', which in turn has a preference for the preposition 'gegen'. It is therefore likely that 'ausräumen' and 'gegen' will often, in fact almost always, occur in the same sentence when they are *not* directly linked. We see here the effects of an indirect lexical preference. In the example case the secondary effect was not critical, since the favored collocation was much stronger, but it clearly shows the limits of unsupervised learning: whenever such a secondary collocation co-occurs with the preposition but without the primary

collocation, it will wrongly attract the preposition anyway. For instance, newspapers routinely give time and location of events with phrases such as these:

"Die CDU wird nach den Worten ihres Generalsekretärs Peter Hintze bei ihrem Kleinen Parteitag **am kommenden Montag in Bonn** eine "Standortbestimmung" vornehmen."
(According to secretary Peter Hintze, the CDU will "reposition" during their local assembly next Monday in Bonn.)
(NEGRA, sentence 19052)

Since the combination weekday+'in' is so common, unsupervised learning will wrongly consider it a strong collocation, when in fact weekdays virtually never bear local modifiers. Luckily, the days of the week are common enough that we have enough instances in our supervised corpus to correctly predict a very low probability for this subordination, but similar effects must be expected to occur with many other recurring phrases. Altogether it is not surprising that our simple model achieves only half of the improvement we have calculated as a theoretical maximum on page 127; it is probable that a more sophisticated predictor could improve on these results. Still, the improvement of 1.5% in attachment accuracy is already a valuable result in itself: it shows that PP attachment information fills a specific gap in the existing language model. Perhaps more importantly, the ratio of sentences that receive an *entirely* correct structure increases from 41% to 47%.

4.5 Learning parse moves

The previous discussion was primarily concerned with uncovering and fixing weaknesses in the model of language that a WCDG defines. To this end, the distribution of typical errors was analysed and additional empirical knowledge about those weak points was incorporated. This approach changes the language model in a way that is overall for the better, even if it can cause deterioration in individual cases.

However, improving the model in this way can not solve the parsing problem entirely, since not all errors that the parser makes are errors of the model; many errors, perhaps the majority, are due to the incompleteness of the solution methods, which may miss the desired analysis of a sentence even if the grammar accurately predicts it. This leads to the question whether statistical methods could not be of use to combat this difficulty as well.

As an example, consider the two analyses of the sentence

"Die Kommission erstellt jährlich einen Bericht über den Stand der Verwirklichung der in Artikel III-209 genannten Ziele sowie über die demographische Lage in der Union."
(The Commission compiles yearly reports about the state of the realization of the

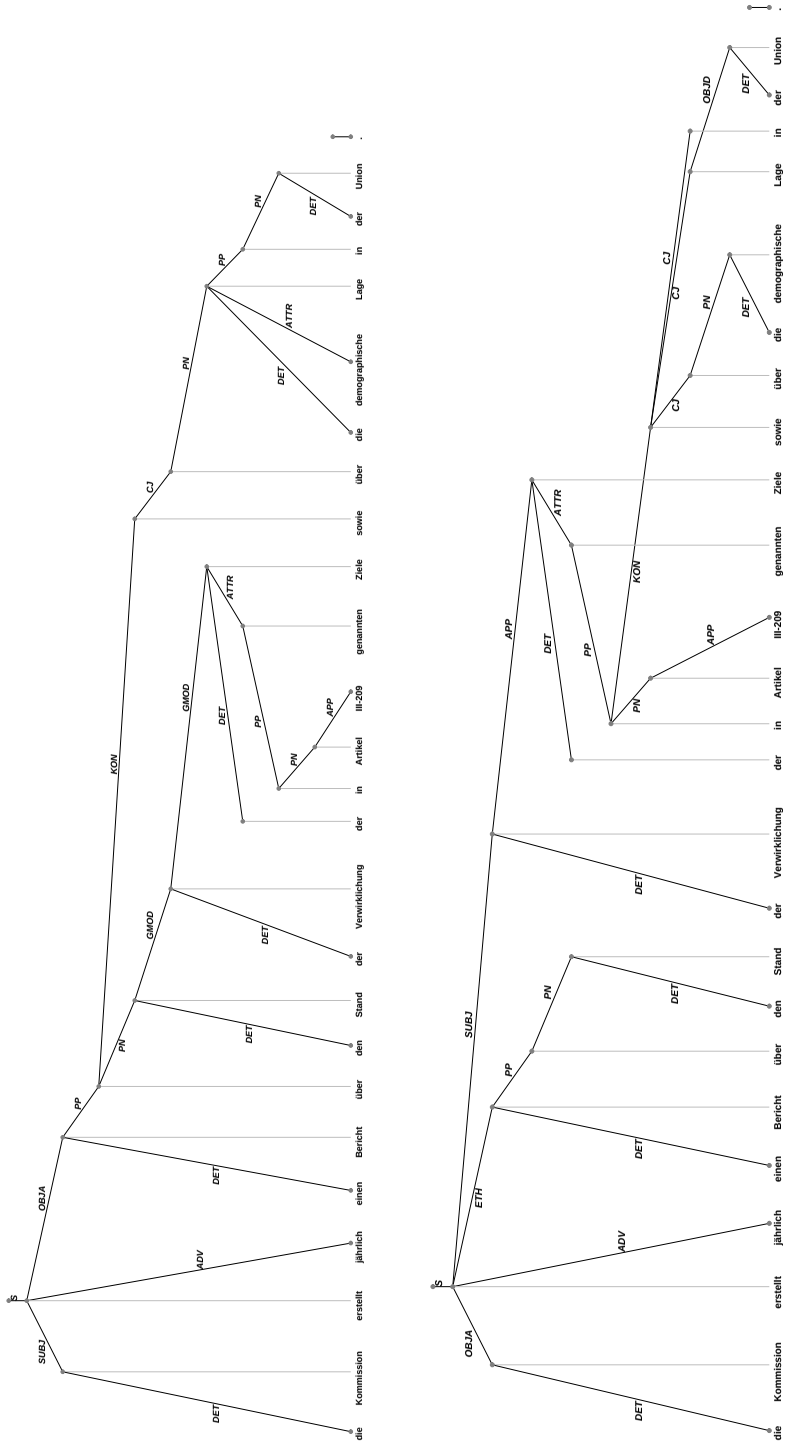

Figure 4.5: Correct and incorrect analyses of a long sentence.

goals named in §III-209, and about the demographic situation in the Union.)
(European Constitution, §III-216)

shown in Figure 4.5. The analysis on the left corresponds to the intuitive meaning, while the right one shows the output of the transformation-based search after a comparatively early interruption. Altogether, this analysis contains seven wrong attachments and ten wrong dependency labels.[10]

The resulting interpretation exhibits a curious mix of correct and incorrect parse decisions. For instance, the parser has correctly analysed the noun phrases "die Kommission", "einen Bericht über den Stand", and even "der Verwirklichung der in Artikel III-209 genannten Ziele". (This supports our earlier reasoning that chunk-internal relations are not a major problem for our parser.) However, the macro structure of the first part of the sentence is wrongly interpreted to mean "The realization gives the commission to a report" rather than "The commission gives a report about the realization". While clearly mistaken, this interpretation is not strictly impossible according to the constraints, and only incurs minor penalties. In contrast, the last words of the sentence are analysed in an obviously wrong way: more than one word is subordinated as a complement of the conjunction 'sowie', which is not allowed by our annotation rules, and in fact causes several hard constraints to fail.

What causes such a variegated parse result? Since the transformation process tries to correct errors in descending order of their severity, it always focuses on the part of a sentence that is currently 'most' wrong, as defined by the set of constraints. If there are several subordinations that violate hard constraints (and initially, there almost always are several, except in trivial sentences), the order in which they are investigated is not well-defined. In this case, the process was stopped before it could ever attend to the errors at the end of the sentence, and the last dependency edges remained as they were after the initial assignment.

From a computational perspective, such behaviour is understandable: if we use an intractable formalism to model the parsing problem, we must live with occasional failures of the resolution algorithm, even if it achieves a competitive overall accuracy. A global search would run the risk of not producing *any* analysis if interrupted too early; and local search algorithms will characteristically attack conflicts only one at a time. But particularly in sentences such as this, which do not contain any unusual constructions at all, it seems inappropriate for a sophisticated syntax parser to make errors of this kind.

In fact, from a linguistic perspective it must be asked whether an uneven analysis such as this is any better than not having an analysis. If we must expect occasional preposterous errors in the output of a system, this severely handicaps its general

[10] Once again, this example is somewhat contrived. The WCDG parser described here does eventually correct all of these mistakes, if it is given enough time to complete the transformation process. In other words, neither of these errors is likely to appear in the evaluations performed in the experiments described in this thesis. Search errors do frequently occur without being corrected, but usually in much longer sentences which are not easily displayed on the printed page.

usefulness because we can never know whether or not this has happened in a particular instance (Kay, 1980). It would be much better if we could at least count on a *plausible* analysis, even if it might not be *correct* in all details, because that is likely not only to contain fewer errors by our evaluation measures, but also to be more useful for processing beyond parsing.

4.5.1 Oracle parsing

Cases such as this demonstrate the penalty we pay for using too general an approach on input that could be analysed just as well (or even better) with much simpler methods. None of the constructions in this sentence is especially uncommon or should require special attention. For instance, the entire sentence contains only 12 different dependency labels (out of the 35 that the grammar defines). Subject and object do in fact occur in this order (and not inverted, as the analysis wrongly predicts), and the non-projective subordination of the co-ordinating conjunction 'sowie', which the grammar allows because it sometimes occurs, is in fact wrong for this sentence. In fact, almost every word of the sentence seems to fulfill exactly the role that a superficial analysis would assume. (We will substantiate this claim in the next section.)

In other words, the parser suffers from allowing alternatives that are necessary for rare cases even when dealing with not-so-rare cases, because it has no way of judging whether a problem instance is pathologically abnormal or run-of-the-mill. Instead it defensively assumes that every sentence is pathological: until it can prove that there is a solution which requires no rare constructions, it will always attempt to instantiate every single one of them. The potential for hybrid systems is obvious here: if we can obtain a simple analyzer that solves some or even all 'easy' problems with little effort, it will be worthwhile to use even if it fails to analyse many difficult ones. For instance, its output could be used to choose a more likely starting point for the transformation process; the deep parser would then not have to spend any time considering alternatives for parts of the sentence that have already received a satisfying analysis. And if the simple parser fails, it will not have lost the deep parser much time; normal processing can simply start from an arbitrary point, as it would have done anyway.

In the example, we would hope that a simple parser would be able to predict at least the noun phrases with much less effort, so that deep analysis could immediately start making the more subtle decisions — for instance, optimally attaching the co-ordination or the final prepositional phrase.

Although a wide range of efficient parsing techniques has been proposed, many of them are dedicated to the construction of phrase-structure trees. While it would be possible to transform phrase structure into dependency structure in order to obtain predictions for the WCDG system, this would somewhat defy the purpose of using

a *simple* parser as an oracle. It would be preferable to use a method that generates dependency trees directly.

Algorithms have also been proposed that operate entirely in dependency space. Hudson (1989) proposes a simple monotonous scheme in which dependency edges are added between suitable words, for as long as this is possible without violating basic tree properties such as projectivity. Covington (1990) omits this restriction and arrives at an even simpler algorithm that can construct any non-cyclic dependency tree (but still prefers continuous constituents). Eisner (1996) proposes a dynamic programming algorithm that constructs the most probable tree from the bottom up in $O(n^3)$ steps, and gives different probability models that can be used with it. Samuelsson and Voutilainen (1997) describe a probabilistic model that comprises independent random variables for dividing input into nuclei, constructing dependencies, assigning edge labels, and determining surface word order. Parsing employs a chart parser that resembles the CYK algorithm commonly used for analyzing context-free languages. Finally, Nivre (2003); Yamada and Matsumoto (2003) propose *shift-reduce* parsing which can guarantee to find a spanning tree for any input in linear time. The basic idea is that an analysis can be characterized by the sequence of moves that a stack-based analyzer would make to construct it, so that a predictor for dependency trees can be constructed by iterating a predictor for individual parse moves.

Some of these approaches appear unsuitable for our goals. In a robust grammar with very broad coverage such as ours, almost any two words can be assumed to modify each other in *some* way; therefore, strategies such as adding an edge whenever this is *possible* are likely to lead to almost random collections of unlikely subordinations which, taken together, make little sense. On the other hand, we would like to avoid elaborate probability models, since we do not want to exert much effort in fine-tuning a model which is not expected to be definitive anyway. Also, for long problem instances even a polynomial algorithm might take up more time than is appropriate for an oracle.

4.5.2 Shift-reduce parsing

The most promising candidate as a fast oracle for parsing decisions seems to be a variant of *shift-reduce parsing*. The characteristic elements of such a parser are an *input pointer* and a *pushdown stack*. The pointer moves monotonically from the first word to the last, and structures can be created that involve the currently visible word and the words or structures on top of the stack. Such mechanisms have traditionally been used for parsing formal languages such as programming languages (Aho et al., 1986). Similar methods were also proposed for parsing natural languages (Marcus, 1980; Shieber, 1983).

The most important difference to parsing formal languages is of course the problem of ambiguity: while formal languages can often be defined so that they allow deterministic analysis, natural languages abound with ambiguity that cannot be resolved

locally, so that more than one parsing action is often possible in a given state. For instance, a *shift-reduce conflict* occurs when it is unclear whether the next word should be included in the constituent under construction or not; in a *reduce-reduce conflict* different constituents can be constructed with the same material. A successful shift-reduce parser must either be prepared to explore both possibilities, or possess very good heuristics for choosing the correct action. Early approaches such as that of Marcus (1980) aimed to explain the human ability for language understanding in linear time, and therefore employed informed heuristics in order to avoid backtracking altogether; with the rise of corpus-based parsing, it became more usual to allow backtracking, but guide it through frequencies obtained from corpus counts.

A shift-reduce parser for constructing dependency structure is closely analogous to one for constructing phrase structure, but since it need only construct word-to-word subordinations, the stack never contains intermediate nodes, but only individual words, and reduce actions only clear the stack without constructing additional structure. We assume a model like the one proposed by Nivre (2003), which allows at most four different moves to be made at any time:

1. SHIFT: The next word in the input string is pushed onto the stack, and the pointer is advanced to the next word.
2. REDUCE: The word on top of the stack is removed.
3. LEFT: The word on top of the stack is subordinated to the next input word and removed from the stack.
4. RIGHT: The next input word is subordinated to the word on top of the stack, and then pushed onto the stack.

Configuration	Move	Edge
[ ] . Prince verkauft sich im Internet	SHIFT	
[Prince] . verkauft sich im Internet	LEFT	Prince $\bullet \xrightarrow{\text{SUBJ}}$ verkauft
[ ] . verkauft sich im Internet	SHIFT	
[verkauft] . sich im Internet	RIGHT	sich $\bullet \xrightarrow{\text{OBJA}}$ verkauft
[verkauft sich] . im Internet	REDUCE	
[verkauft] . im Internet	RIGHT	im $\bullet \xrightarrow{\text{PP}}$ verkauft
[verkauft im] . Internet	RIGHT	Internet $\bullet \xrightarrow{\text{PN}}$ im
[verkauft im Internet] . $	REDUCE	
[verkauft im] . $	REDUCE	
[verkauft] . $	REDUCE	verkauft $\bullet \xrightarrow{\text{S}}$ null

Figure 4.6: Analysing a simple sentence with shift-reduce parsing.

For any connected, projective dependency tree over a sentence of n words, there is a unique sequence of at most $2n$ parse moves that will construct this tree. The key

idea is to shift tokens onto the stack until the top word and the input word form a left or right subordination, and reduce a token only when it has already acquired all right subordinations that it should have. Figure 4.6 shows the sequence of parse moves necessary to correctly analyse the sentence

"Prince verkauft sich im Internet"
(Prince sells himself on the internet)
(heiseticker, item 15199)

Square brackets denote the parse stack, with the top word at the left, while the stop symbolizes the input pointer. Note that parsing takes exactly $2n$ steps.

Various variants or extensions of this algorithm are possible that change the set of trees that can be constructed. For instance, the example shows how labelled dependency trees can be constructed by set multiplying a set of permissible labels with the set of parser moves. Also, the REDUCE move is usually restricted to parser configurations in which the top stack word already has a regent. If this condition is lifted, a forest of unconnected trees can be generated. We do in fact allow precisely this, for instance to allow the final full stop to form a second null dependency.

It is to be expected that the utility of an attachment predictor would correlate with the accuracy of its predictions. In the case of shift-reduce parsing, this will depend on how well an oracle is able to predict the correct parse move to be made from the information it has available. In our favour is the property that the problem has a constant, small branching factor: for unlabelled trees, the decision to be made at each step is among a set of not more than four alternatives (since not every move is even possible in every parser configuration). It might be possible to construct a predictor that archives an accuracy high enough that it can be run with no backtracking at all, and still produce an analysis that is accurate enough to be useful as an oracle.

4.5.3 Previous work on deterministic dependency parsing

Several previous works have reported on constructing just this sort of *deterministic* shift-reduce parser. Nivre (2003) describes a rule-based parser that creates unlabelled, connected syntax trees, and compares the impact of different scheduling policies on the accuracy of parsing a small toy corpus. The simplest possible policy is a static ordering on the possible moves, so that of all possible moves, the highest-ranked one is always taken. For instance, parse moves could be taken in the preference LEFT-RIGHT-REDUCE-SHIFT.

Some simple extensions are also described that help choose between particular alternatives in a more reasonable way. For instance, a SHIFT/RIGHT conflict arises when the input word could be a post-modifier of the stack word, but also a pre-modifier of the word after it. In this situation, the SHIFT action is preferred if the stack word is a verb, which corresponds to a preference for the pre-modifier analysis.

The best of these parsers achieves an attachment accuracy of 89.0% on a corpus of about 4,000 words of Swedish.

The parser described by Yamada and Matsumoto (2003) works with a similar set of moves, but can make several passes over the input so that its worst-case time complexity is quadratic rather than linear. Its scheduling policy is learnt from a treebank; the relevant features that determine its decisions are not only the words to the left and right of the input pointer and their syntactic categories, but also the words and categories that are already subordinated to the hypothetical new regent. For instance, in the phrase

"the buyers and sellers of last resort who were criticized after the 1987 crash"

the word 'who' should attach to the regent 'sellers'. Their parser is able to detect that the word 'were', which modifies 'who', indicates the plural reading of 'who', and therefore the plural regent 'sellers' is more appropriate than the singular 'resort'. To handle the very large space of possible features defined by this model, the system uses *support vector machines*, which are able to generalize efficiently even in many dimensions and at the same time avoid overfitting. With the best parameter settings, a structural accuracy of 90.3% is achieved when analysing the standard part of the Wall Street Journal corpus.

Nivre et al. (2004) likewise extend the rule-based shift-reduce parser so that it assigns dependency labels as well as structure, and learns its decisions from a treebank. The memory-based ID1 algorithm is chosen for its flexibility in dealing with unknown parser configurations. This parser was later also applied to the Wall Street Journal corpus (Nivre and Scholz, 2004) and there achieved a structural accuracy slightly below previously published results, but a competitive labeled accuracy.

4.5.4 A simple parser of German

A conclusion of the work cited last was that the accuracy of shift-reduce parsing degrades gracefully with increasing sentence length. Because it does not open the full space of theoretically possible word-to-word modification that WCDG or even a PCFG parser must allow, it misses some of the unusual constructions, but also manages to construct the predominant easy cases quickly, without being distracted by pathological alternatives.

Our corpus of German sentences annotated with dependency structure should provide ample material for constructing a fast oracle parser for WCDG. To implement a simple deterministic data-driven shift-reduce parser, it remains only to design its parameters more exactly. The input to this parser should consist in the words of a sentence with disambiguated part-of-speech tags. This allows us to try out both lexicalized and unlexicalized models and test their accuracy. The output of the simple parser should be a syntax tree describing the input and, optionally, dependency labels. A labelled dependency structure is more informative for guiding WCDG,

but since dealing with labelled dependency trees effectively decreases the size of the training set, it might be preferable not to predict edge labels. The other tasks of the WCDG parser will be ignored: the oracle will not construct reference relations, since they are rather rare, and where they occur they can usually be reconstructed accurately after the fact.[11] Also, it will not try to disambiguate the morphological features of homonymous word forms.

For simplicity, the probabilistic model used to make parsing decisions will consist of simple tuple counting, where the action that was observed most often in the current parser configuration is always chosen. If the current configuration was never observed, the parser will back off to a more general description of the current configuration until observations are available, or finally use SHIFT or REDUCE as appropriate if no data are available at all.

A fundamental limitation of the chosen model is that the various cases of controlled non-projectivity that our model of German allows cannot be constructed at all: One of the two words that a nonprojective dependency connects will always be either be already reduced away or not yet accessible in the input string when it would be necessary. Entirely different moves would be required to allow such dependency edges to be constructed, and it would defy the basic principle that only the two "visible" words can ever be subordinated. We work around this problem by applying a primitive normalization to the training set: all non-projective subordinations in the training set (for instance, most relative clauses) are simply changed to form fragments instead, and the predictor is trained on the normalized corpus. This changes about 1% of all syntactic dependencies, but since we do not expect a dependency accuracy anywhere near 99% in the first place, this is a reasonable compromise that keeps the oracle parser simple while not distorting its output too much. This is confirmed by previous research on Czech (McDonald et al., 2005); this language contains even more non-projective dependencies than German, but a parser that acknowledges non-projectivity explicitly still improves accuracy only by about 1% over one that does not.

Another fundamental weakness of the algorithm is that it is not generally possible to predict the correct move entirely from the local context, as the description above implied. Often the decision whether to reduce a word or not depends on whether or not there are words beyond the current input word that should eventually modify the word on the stack. It is clear that this distance can in principle be unlimited, so that the feature space can grow very large indeed. We do not pursue this way further, since we are not interested in statistical parsing for its own sake, but only as a *fast* helper program. And indeed the oracle parser can run very fast, since it only has to perform at most two table lookups for every word.[12]

[11] This is of course an artifact of our restricted view of reference, which includes only relative pronouns. A *general* treatment of reference would be more and not less difficult than syntax parsing.

[12] Our implementation in fact parses the entire test set in 40 seconds.

It remains to determine the exact set of features that will be used to define our similarity model for parser configurations. We will choose features from the following possibilities:

1. 'Top': The word on top of the stack, characterized by its reading or its part-of-speech tag.
2. 'Context': A limited representation of the dependencies that have already been constructed for this word. This is a three-bit number representing whether or not the word already has a left dependent, has a right dependent, and is itself subordinated.
3. 'Distance': The linear distance between the top word and the input word.
4. 'Lookahead': The part-of-speech tag of the input word and its successors.

The set of this features was intentionally chosen to be somewhat extensible (via a variable lookahead window and optional lexicalisation) but, overall, simple. While a richer feature set would certainly allow better decisions to be made in some cases, it would require much more experimentation in order to obtain the optimal setting of all possible parameters; in fact, feature-rich approaches often report that an exhaustive tuning was not possible, so that some of the theoretical advantage may not even be reaped. Our goal is not to perfect deterministic parsing, but only to measure its utility for full parsing; therefore we limit ourselves to finding the settings that perform best with this restricted model.

For best comparability, we use the same section from the NEGRA corpus as before as our test set and all other available dependency trees as the training set. It turns out that the best combination of features for this task is to use the distance and context features, two words of lookahead, and no lexicalisation. With these settings, the structural and labelled accuracy on these sentences turns out to be, respectively, 85.0% and 80.7%. This is quite a respectable number even when compared with the results for statistical parsers reported above, and with the WCDG parser itself, which achieves 89.3% and 87.5%, respectively.

The result confirms the uneven distribution of phenomena that was to be expected: the 'easy' attachments (those that the oracle can get right reliably) are rather common, and those that require deeper analysis are comparatively rare. For instance, the shift-reduce parser will invariably predict that every article it encounters modifies the next noun encountered. Although German allows some types of noun phrases to contain other embedded noun phrases (see p. 113), instances of this phenomenon are very rare, and therefore almost all of these predictions are indeed correct. Since determiner relations typically account for about 10% of all dependencies in written German, this alone ensures an easy 10% of accuracy for the oracle parser. In contrast, the WCDG parser has to check each case carefully for whether it might be one of the rare ones, and in fact it is occasionally fooled into assuming that it is.

As an example of how far 'easy' decisions can go towards full parsing, consider again the sentence pictured in Figure 4.5. In Section 4.5 we claimed that this sentence was altogether normal and should require no special attention to analyse. We can now back this assertion up by submitting the sentence to the part-of-speech tagger and to the shift-reduce parser. And in fact it turns out to be analysed without any errors, both structurally and with respect to its labels; in other words, the oracle parser actually produces the *exact* analysis displayed on the left in this picture.

This, of course, is not representative of the accuracy of the oracle in general; this sentence was chosen especially for exposition because it not only appears easy, but actually *is* easy. Overall, only 15% of all sentences actually receive analyses without any errors. Sentences of comparable length are generally analysed with a labelled dependency accuracy that is close to 80% rather than 100%. Nevertheless this demonstrates the advantages that using such an oracle as a starting point could bring with it: in the optimal case, WCDG will reach the correct solution almost instantaneously rather than after a long search. It may still spend a long time verifying that it really cannot find a better analysis, but if interrupted during that time, it will then return the correct result rather than an approximation. In the more likely case that the oracle is somewhat imperfect, the parser could still profit from having many of the easy problems already solved, so that effort can be directed at those problems that actually need solving.

It bears repeating that we have certainly not exhausted the possibilities of data-driven shift-reduce dependency parsing. It is well-known that low-frequency events are important for the last few percent of accuracy that can be extracted from a given training set (Collins and Brooks, 1995), and that adaptive methods for feature selection could be used to incorporate them; in fact, both Nivre and Matsumoto do precisely this. Some examples of errors that a more sophisticated shift-reduce parser could probably avoid are shown in Figure 4.7.

In the first tree (NEGRA, sentence 18613), two noun phrases have been analysed as subjects to the same verb, although this is clearly wrong, and never occurs in the training set. Extending the 'Context' feature of the similarity model to consider the labels of dependencies already established could prevent this error, but might also exacerbate the problems of sparse training data (and in fact, adding such capability to our oracle parser does not increase its own accuracy). In the second tree (NEGRA, sentence 18635), the subject 'Doppelzüngigkeit' of the finite verb 'ist' has been analyzed as a genitive modifier, which is possible but very unlikely, since its determiner 'diese' is not marked as a genitive. This clue might be picked up by a corner-word feature like the one used by Yamada and Matsumoto (2003). The second clue (the fact that the finite verb lacks a subject) is harder to exploit, since at the time the decision must be made, the parser cannot know whether the verb might not yet acquire a different subject before it is finally reduced. Finally, the third tree (NEGRA, sentence 18678) shows an error caused by our heavy-handed normalization of the training set, where non-projective dependencies were simply assumed to be frag-

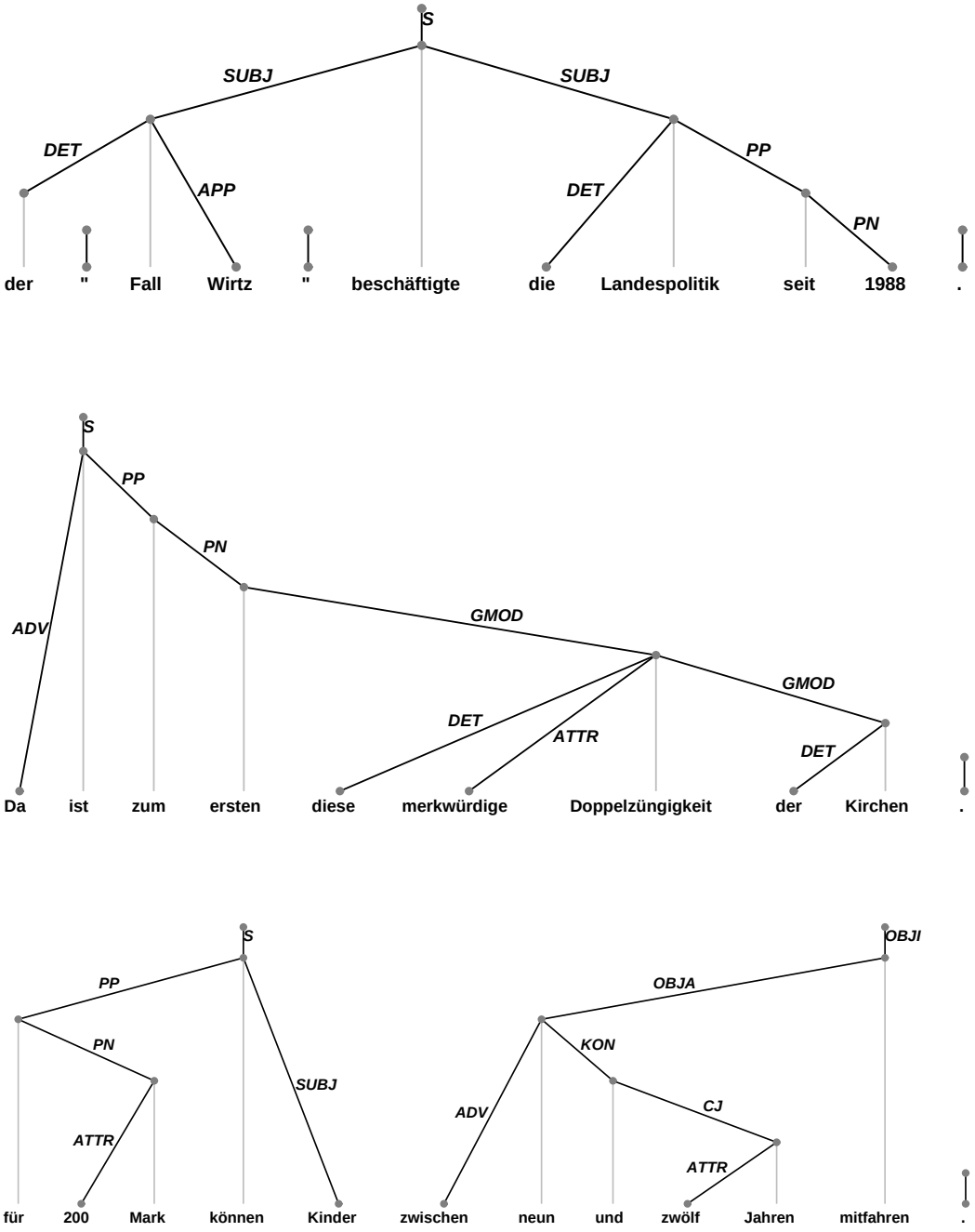

Figure 4.7: Errors of the oracle parser due to simplifying assumptions.

ments. Since OBJI (subordinated infinitive) dependencies are non-projective more often than not, they are usually normalized, and a null subordination is actually the event most often seen in the training corpus. Nivre and Nilsson (2005) solve this problem by normalizing the training set to be entirely projective, but encoding the non-projective structure in special dependency labels instead, and then denormalizing the parser output again. Other approaches (Levy and Manning, 2004; Hall and Novák, 2005) instead suggest corrective models that can reconstruct non-projective relations from the output of normal constituency-based parsers, by investigating the neighbours of the falsely suggested regent.

Despite such clear imperfections, our simple oracle parser could be sufficient to establish the fact that parser/parser hybrids can yield a significant benefit for WCDG parsing. Since the oracle can easily be replaced when more accurate models become available, our system will still be able to benefit from future research even when it is done by others.

4.5.5 Oracle parsing for WCDG

Shift-reduce parsing can be used as a predictor in much the same way as the other knowledge sources discussed above. It receives both the input string and the preferred part-of-speech for each word, constructs its syntax hypothesis, and reports its recommendations in the form of a suggested regent and a suggested label for each word in the input. The only question is to what use these suggestions should be put. We could simply use them for selecting the initial structure that is then subjected to normal heuristic transformation as necessary. This would limit the influence of the oracle to an initial suggestion that WCDG can override at no cost. To receive some guidance even during transformation, we could also encode its predictions into constraints which express that WCDG should follow the suggestions of the oracle. Such integration constraints can be written as follows:

```
{X:SYN} : 'SR:regent' :
  ~root(X^id)
  ->
  predict(X@id, SR, gov) = X^to;

{X:SYN} : 'SR:null :
  root(X^id)
  ->
  predict(X@id, SR, gov) = 0;

{X:SYN} : 'SR:Label' :
  predict(X@id, SR, lab) = X.label;
```

A related question is once again which penalty to choose for such constraints to ensure that WCDG receives guidance from the oracle, but is free to disagree if necessary. This necessity is particularly clear for an oracle that, unlike any described so far, is a *full* parser; requiring the main parser to adhere to the decisions of the oracle would make it quite pointless to run WCDG at all, since it could not improve on the results of the oracle.

With the right penalty we can ensure both an informed starting point and a continuous guidance for the transformation process. Since the initial analysis is found by selecting the subordinations with the highest unary score, the constraints shown above can ensure that transformation starts out with the predicted analysis with no change to the program itself. They also provide an incentive for the algorithm not to stray too far from that analysis.

As with previous oracles, we first test a value of 0.9 as the constraint penalty. This ensures that the oracle takes precedence over default and most preference rules, so that usually the initial analysis will indeed be the predicted one. However, it can be overridden by more important rules such as morphosyntactic agreement. Recall that the most successful shift-reduce model is entirely unlexicalized, so that it can not use agreement rules to detect misleading category sequences; but since agreement is encoded by more important rules, WCDG can overrule such improbable subordinations easily. For instance, the misleading example cited on p. 106 as an instance of a typical misleading category sequence is indeed analysed by the oracle parser as if it contained the noun phrase "das Kostensenkungen"; but WCDG can immediately detect that this would violate the concord of number, and therefore override the structure suggestion before parsing even begins.

	Accuracy	
Penalty	structural	labelled
0.3	89.9%	88.0%
0.5	90.4%	88.5%
0.7	90.6%	88.8%
0.9	91.4%	89.7%
0.95	91.5%	89.8%
1.0	89.3%	87.5%

Table 4.15: Parsing results with shift-reduce oracle.

Table 4.15 shows the results of an experiment with different penalties for the integration constraints (a value of 1.0 makes them inoperative, so the last line simply repeats the figures from Table 4.8). It can be seen that the oracle parser is very useful indeed: it can reduce the attachment error rate by up to 20%, raising structural accuracy well above 91%.

This result is remarkable because we know that the oracle itself is considerably less exact than the WCDG parser alone, so the synergy between the two different

parsers must be considerable. We can attempt to measure this in a more direct way by regarding the individual attachment errors that either component makes. It turns out that of the 2517 attachment errors made by the oracle parser, 1027 were also made by WCDG, but 1490 are not. At the same time it correctly attaches 770 words that WCDG handles wrongly.

Another advantage of having a complete parser as an oracle was briefly mentioned above: it could improve the time-adaptive behaviour of parsing considerably. We showed how in the ideal case, the correct analysis might be found almost immediately instead of after a long transformation process. That, of course, only holds if the oracle parser makes no mistakes. We are now in a position to test whether similar improvements can be obtained even with an imperfect oracle parser. To the end, we repeat both the experiments from Section 4.1.8 (WCDG with hybrid part-of-speech tagging) and Section 4.5.5 with a shorter timeout. In the extreme case we allow no time for the transformation algorithm whatsoever, so that the analysis that the parser returns is determined solely by the unary constraints, which can be evaluated in linear time. With successively longer periods of time allowed for optimizing, we would expect the average parsing quality to rise monotonically.

Time limit	with oracle	without oracle
0s	81.9%/72.0%	56.5%/49.6%
20s	88.0%/85.5%	77.6%/74.7%
100s	86.5%/84.3%	86.3%/84.2%
200s	90.8%/88.9%	87.9%/85.9%
1000s	91.5%/89.8%	89.3%/87.5%

Table 4.16: Parsing under time pressure with and without oracle parser.

Table 4.16 shows the results of simulating time pressure in this way (again, the last line repeats the figures of previous experiments). As expected, the parser generally does better the more time it is allowed to spend, although due to the rather small test set size the increase in accuracy is not entirely monotonic. The more important result is that the oracle parser not only raises the maximal accuracy achieved, but also decreases the time required to obtain a given level of accuracy. For instance, if a maximum of 20s is allowed for parsing a sentence, the average accuracy reaches a level that WCDG normally achieves only after 200s. The greatest advantage can be seen in the edge case where no time is allowed for transformation at all: while WCDG barely manages to achieve an accuracy of 50% (that is, to produce more correct than wrong attachments) when only the unary constraints are used, the hybrid parser already starts out with a respectable accuracy. (Note that these figures are actually lower than the accuracy of the oracle parser in isolation; this shows that the unary constraints overruling some of the oracle's decisions are in fact not always correct in doing so.)

This experiment confirms one of the key promises of hybrid systems: a careful combination of components with different performance profiles can create a whole that profits from the advantages of both parts.

4.6 Combination of different predictors

So far we have only investigated the effect that one particular stochastic knowledge source has on our parser; this allowed us to precisely measure the benefit that each of them brings. However, we used the part-of-speech tagger described in Section 4.1.7 in all later experiments, since it was instrumental to parsing general text in the first place. In this respect we have already shown that more than one external model can profitably be used at the same time.

It remains to investigate the effect of combining the more complicated models with each other as well as with the normal grammar and part-of-speech tagger. For several reasons, it is not to be expected that the benefit of two additional sources will simply add up. The value of an external predictor to WCDG lies in its ability to predict subordinations which are correct but which the WCDG parser alone did not find. If the predictions that two different language models make about a given attachment are both correct, they may constitute even stronger evidence for that subordination than either alone, and may make it even more likely that it is eventually selected, but it can be established once only. In other words, a WCDG error that was corrected by an ancillary component cannot be corrected a second time, but it may very well be reintroduced by misleading evidence from a second predictor. Only if the sets of errors that each predictor corrects were entirely disjoint could we expect the optimal result that all error rate reductions simply add up.

Nevertheless, the different methods and training data employed by some predictors may lead to at least some synergy. For instance, we would expect that supertagging and PP attachment work well together, since preposition attachment is one of the structural relations that supertags do not determine (Nasr and Rambow, 2004). On the other hand, chunk parsing predicts component boundaries that are also part of the output of a shift-reduce parser, so that little additional benefit is to be expected.

Table 4.17 repeats the results of previous experiments for comparison, and adds results of further parsing runs in which more than two predictors were employed. All experiments marked 'POS' employed the hybrid part-of-speech tagger described in Section 4.1.8; chunk parsing (CP), supertagging (ST), PP attachment (PP) and shift-reduce parsing (SR) were also used exactly as in the previous experiments, but now in combination with each other. The last experiment employed all five statistical predictors.

This comparison shows that a further benefit can be obtained by combining more than two external predictor with the base grammar, but that the benefit falls far

		Accuracy	
Experiment	Predictors	structural	labelled
1	none	72.6%	68.3%
2	TnT	89.0%	87.1%
3	hybrid POS	89.3%	87.5%
4	POS+CP	89.8%	88.0%
5	POS+ST	91.9%	90.5%
6	POS+PP	90.6%	88.9%
7	POS+SR	91.5%	89.8%
8	POS+PP+SR	91.4%	89.6%
9	POS+PP+ST	92.0%	90.6%
10	POS+ST+SR	92.2%	90.7%
11	all five	92.3%	90.9%

Table 4.17: Overview of parsing results with and without predictors.

short of the theoretical maximum. A particularly unsuitable combination is that of PP attachment with shift-reduce parsing (experiment 8). This combined experiment results in an accuracy between that of the individual experiments 6 and 7, that is, the parser actually loses some accuracy when adding the PP predictor on top of the oracle parser. This might be because of an ill-designed duplication of work: the oracle parser predicts the regents of all prepositions in the same way as all other subordinations, without regard to lexical identity (recall that unlexicalized shift-reduce parsing resulted in the best overall accuracy of the oracle itself). The PP attacher also predicts the regents of all prepositions, but makes crucial use of lexical identity to prefer plausible over merely close combinations. Apparently the predictions about prepositions are at odds often enough that the parser wastes much effort on trying to reconcile the incompatible, and so misses a few more correct attachments.[13]

On the other hand, PP attachment and supertagging do show a small joint benefit over their individual contributions, while the combination of supertagging and oracle parsing improves significantly upon either alone, and combining all five predictors yields the highest benefit. In particular, for this particular corpus, multiple predictors are the only constellation which improves even the *labelled* accuracy above 90%.

[13]It turns out that this is indeed the case; by exempting prepositions from the shift-reduce constraint we can increase the structural accuracy to 91.7%, which is above both previous experiments.

Chapter 5

Conclusions

5.1 Summary

We have presented a wide-coverage hybrid parsing system for written modern German. A set of declarative rules formulated as WCDG constraints, in combination with heuristic knowledge extracted from annotated dependency corpora, together define an algorithm that achieves competitive accuracy when analyzing German texts from various sources and domains. This success was made possible by the combined properties of both paradigms of natural language processing.

The rule set in itself describes many phenomena in great linguistic detail and can be used to deduce the preferred interpretation of language utterances in an informed way (see Section 3.4.1). From a theoretical point of view, it is not remarkable that this should be possible; WCDG was intentionally conceived as a formalism that can express *any* preference that a linguist might declare in comparing possible interpretations of an utterance (because there is no limit to the expressiveness of operators that one can add to its constraint language). For this reason it *must* be able to support any rule that a linguist might propose, as long as it can be formally expressed at all. The novel result of this thesis, then, is not that a WCDG can describe an entire natural language well, but merely that this is possible with a feasible amount of work.

In comparison with previous work (Schröder, 2002), this work represents the first WCDG of a natural language that is intended to be *comprehensive*. For instance, it can process all important sentence types, deal with unknown words in an adequate way, and generally analyse texts from different domains and text types indifferently, as evidenced by the comparable results when parsing standard corpora and a variety of dependency corpora collected during the development of WCDG itself.

In comparison with the grammars cited in Menzel and Schröder (1999); Schröder (2002), from which it indirectly derives, the present rule set has been considerably

extended in various respects. Perhaps most obviously, the set of syntactic dependency labels has been greatly extended (see Section 3.2.3). At least in this regard the grammar is intended to be essentially feature-complete, i.e. the introduction of even more labels should not be necessary in order to represent any utterance from standard modern German. The set of covered syntactic constructions, while certainly not complete, nevertheless allows the great majority of utterances encountered so far to be represented accurately. The limits of the grammar in this respect are often those phenomena that pose fundamental problems for dependency analysis; for instance, an adequate treatment of complex elliptical co-ordination clusters would be possible only through the use of empty elements or transformational analysis, which the WCDG reference implementation currently does not support.

To this end, many existing rules were changed in order to allow additional configurations of dependency edges needed to cover existing phenomena. Many more rules were also added in order to describe newly introduced relation types, or to contain the possibilities enabled by other changes. The set of operators in the CDG constraint language was also extended in comparison to previous work. In particular, Schröder (2002) noted that the restriction of WCDG to unary and binary constraints in practice severely limits the constraint writer, and proposes using constraints of higher orders. This suggestion has been partially followed with the invention of *context-sensitive* operators such as `is()` and `has()`, which allow non-local conditions to be checked as long as the triggering configuration itself consists of not more then two dependency links (Foth, to appear).

Although it is capable of making fine distinctions, on its own the WCDG grammar achieves substandard performance when applied to raw text. The overt cause for this the intrinsic complexity of WCDG: the optimization problems that it defines are infeasible to compute exactly, and because the grammar aims for extremely wide coverage, it increases the size and difficulty of actual parsing problem instances even further. But there is another important reason: our rules mainly describe the *possibilities* of syntactic expression, and not the *probabilities*, which are much more difficult to enumerate. Although many defeasible constraints exist which allow but disprefer certain constructions, there are many more possible but implausible constructions that are not dispreferred, simply because it is difficult to conceive of them before they are computed by a machine parser. Those preferences that do exist are mostly confined to rules about label and category combinations, while the equally important lexical identity of words is mostly ignored. Again, this is because it is infeasible to write rules about every noun or verb of a language by hand.

This sort of problem is typical for hand-written grammar formalisms, which depend on foreseeing every combination that might arise in parsing before parsing is begun. Therefore, we borrowed results from the statistical paradigm. Methods have been proposed for adding data-driven components as oracles to the full parser. They have proven to be useful for reducing both modelling and search errors. The work on solving these subtasks in this thesis is not in any way original; in fact, existing

programs were reused or retrained where possible, or else the simplest conceivable implementation of an oracle was chosen. Thus, the accuracy of our oracles falls below the state of the art in every case. Nevertheless, their inclusion proved valuable to the parser in every case although the oracles were less than perfect. In fact, this even strengthens the case for hybrid systems, since improving the oracles will hardly decrease the accuracy of the entire system.

The synergy achieved in the hybrid system is evidenced by the fact that it delivers better results than with only the hand-written grammar (and in the case of full dependency parsing, better than the oracle parser alone). From the perspective of writing broad-coverage natural language analysis systems, the benefits are varied:

- Statistical methods can *make very broad-coverage analysis feasible* in the first place. We have seen how a rule set that tries to allow every conceivable interpretation of a sentence becomes overwhelmed with implausible alternatives (Section 4.1.1), so that no useful results can be obtained in practice even though they may be defined theoretically. Even an oracle as simple as trigram category prediction suffices to overcome this obstacle (Section 4.1.8); almost the same level of accuracy can be reached as in the idealized setup where syntactic categories are presumed known. Category prediction fulfills the role of an *enabling* technique for realistic WCDG parsing.

- Statistical methods can *help arbitrate between expending effort on rule-writing and on data collection.* We have seen (Section 4.1.1) that very many errors due to category ambiguity are essentially identical, and that one could conceivably write more rules that disprefer categories in the particular context of other categories. In fact, a successful part-of-speech tagger has been written which does just that (Brill, 1995). However, that work selected its rules automatically based on their improvement to a given corpus. It would certainly be far more effort to choose the most important rules and formulate them as WCDG constraints, or even to test the effect of *all* such conceivable constraints by repeated parsing of a corpus. Instead, the output of an existing component can be trivially integrated with a single constraint and used to guide parsing without individual human judgement. The only remaining difficulty is to obtain tagged training data where they are not already available, a task that is easier than full annotation.

- Statistical methods can *be supplemented by judicious rule-writing* so that both the oracle and its effect on the parser are improved. Where the results of the part-of-speech tagger contain systematic errors, whether due to bad training data or to limitations of the probability model, individual rules can be written *after* the fact to correct those recurring errors that have proved to hinder parsing particularly (Section 4.1.7). In this way, the probability model can be made to do most of the work. Writing correction rules as part of the external oracle is more convenient than having to formulate them as declarative WCDG

constraints. It also allows a clear separation of the co-operating components: workarounds for bugs in the part-of-speech tagger can be kept in the oracle component rather than the grammar proper, which is conceptually independent of the oracle.

- Statistical methods can *selectively refine the language model* by machine learning (Section 4.4.5). They can be used in a suitable intermediary type of rule, when hard rules are too inconclusive and general preferences are too unreliable to be workable. In particular, statistical methods can retrieve the *lexicalized* preferences that are crucial for many instances of real-life ambiguity resolution better than hand-written rules could. This use of corpus data does not merely duplicate knowledge already implicit in the grammar itself to be faster accessible, it introduces genuinely new information.

- Statistical methods can also *guide the parsing process* so as to achieve better results (Section 4.5.5), even if their contributed knowledge is not strictly new to the parser. A part-of-speech tagger will ultimately suggest the verb or noun reading of a word for fundamentally the same reason that a full analysis would: because it fits the utterance context better than the alternative. But by making this knowledge quickly available, it is possible to guide the parser quickly to the solution that was always possible, but difficult to find because of the multitude of alternatives. An overall benefit ensues even though the oracle makes some mistakes, largely because of the ability of WCDG to integrate uncertain knowledge through defeasible constraints. It is easy to measure what the maximal benefit of such an oracle would be, and it can be shown that this limit can be approximated reasonably well by empirical means.

- Statistical methods can also *shift the time-dependent performance profile* of parsing towards a more efficient behaviour. The integration of oracle predictions as additional constraints superficially seems to increase parsing effort, since it increases the number of rules to be evaluated in every step, and also causes new conflicts that trigger more transformation steps. Nevertheless, the overall effect is a speed-up rather than a slowdown: constraint parsing fundamentally involves a trade-off between processing time and quality of results, and speeding up parsing means that the same quality can be achieved in less time. In this sense the effect is a result of the previous point. In the case of a fast and simple dependency parser used as an oracle, the speed-up for achieving a comparable parsing accuracy can be as much as an order of magnitude (Section 4.5.5). This shows that the different algorithmic properties of partial and full parsing combine usefully into a hybrid analyzer.

It should be stressed that all of these results apply to the case where statistical methods are used in a supplementary role, as part of a hybrid system that is to a large degree axiomatic in nature. Chomsky's critique of n-gram techniques as *complete* models of natural language is as pertinent as it was in 1957; but used as helpers

in an informed way, and integrated via defeasible constraints, they can nevertheless contribute to automatic parsing, by integrating the empirical perspective on natural language use.

5.2 Future research

We have shown that various different models of statistical knowledge can increase the accuracy of analysis when their predictions are taken into account by the WCDG parser. In many cases we did not spend as much effort as possible on refining these statistical models, since one of our motivations was to *avoid* manual fine-tuning, whether of grammar rules or of probability computations. Therefore, as noted before, many of the predictors we have implemented could profit from further research. At the same time, we aimed to integrate their predictions in a way that allows the easy exchange of one predictor for another, so that improvements can be exploited even when they result from independent research. New methods of statistical natural language processing are still being invented; as predictors with better performance become available, it will be possible to harness them by simply replacing the predictor. Supertagging, chunk parsing, or PP attachment seem like probable candidates for future improvement.

It is also probable that statistical predictions could prove useful in other domains than the five presented here. For instance, recall that adverb attachment causes about half as many problems for our parser as preposition attachment (see p. 127). The latter problem was tackled both for better comparison with previous work, and because it promised greater immediate benefits to parsing accuracy; however, it seems likely that lexicalized counts could also be of use for the attachment of adverbial modifiers. The existing predictor programs are written with the assumption that they deal with prepositional phrases, and therefore cannot quite be reused for the related task, but similar predictors could be written that use corpus counts of adverbs and their regents.

Similar effects can be expected at least with some types of nominal attachments such as subjects, objects, and adverbial nominal modifiers. In many cases, the preferred interpretation of a sentences seems to result not from syntax or morphology, but from general assumptions about the typical actors in well-known contexts. Consider the following examples, in which seemingly identical syntactic structures should be interpreted differently for semantic reasons:

"Die Regulierungsbehörde wollte diese Vorwürfe nicht kommentieren."
(The regulatory authority declined to comment on these allegations.)
(heiseticker, item 3045)

"Ein fertiges Produkt kann das Unternehmen aus Palo Alto indes noch nicht vorweisen."

(However, the Palo Alto-based company cannot exhibit a complete product yet.)
(heiseticker, item 6814)

Because nominative and accusative forms coincide for all forms in these sentences, morphology does not offer a clue to distinguish subjects from objects. But nevertheless these two utterances are clearly instances of two different sentence types, even when taken out of context: authorities usually comment on allegations, but products are typically presented *by* companies, while the opposite is unheard of. This shows that lexicalized knowledge can also be of help in disambiguating subjects from objects.

The same goes for the distinction between nominal modifiers and objects:

"Das bestätigte ein Sprecher **der Behörde** am Samstag."
(A spokesman for the agency confirmed this on Saturday.)
(constructed)

"Das bestätigte ein Sprecher **den Journalisten** am Samstag."
(A spokesman confirmed this to the reporters on Saturday.)
(constructed)

Since spokespeople are commonly employed *by* organisations, but converse *with* reporters, a parser should analyze the first highlighted phrase as a genitive modifier, but the second as an indirect object. Again, German morphology can often resolve such ambiguities, since genitives differ systematically from datives; but in both examples the two forms coincide, so that *only* world knowledge, or at least some approximation of it, can make the decision.

Finally, there are also cases in which morphology is unhelpful despite overt case marking:

"Die Massenproduktion läuft **Ende 2002** an."
(Mass production will start at the end of 2002.)
(constructed)

"Das Schiff läuft **einen neutralen Hafen** an."
(The ship calls at a neutral port.)
(constructed)

Here the parser must recognize that the first highlighted phrase is a temporal modifier, while the second is a direct object, although both are overt accusatives. In combination with an optionally transitive verb, once again a pair of utterances results that can only be treated correctly with the awareness that a year is a good temporal modifier, but a harbour makes a better destination for ships.

At least two difficulties can quickly be recognized concerning the automatic acquisition of such knowledge. First, the appropriate level of representation for such preferences is probably not the lexeme. Although "Hafen" *(port)* is clearly the most typical destination for a ship, many related terms could be used almost as well, and

year numbers are virtually an unlimited set. In the same vein, it is not only public agencies who employ spokespeople, reporters can just as well be described as 'journalists', etc. To profit from the type of knowledge required here could necessitate a classification of words by *semantic closeness*, which so far is almost entirely absent from our grammar; in other words, it would require the addition of an *ontological hierarchy* to the lexicon.

A second problem is that the distinctions that we want to make are difficult to extract from text in large quantities: we can observe that "Schiff" and "Hafen" occur in the same sentence more often than by chance, but we cannot easily infer which of them is the subject. Essentially, the sentence would have to be parsed before the correct count could be established. This poses a serious problem for data acquisition, since learning such actor preferences is likely to require quite as much training material as preposition attachment. The problem might be solved with an additional component to the parser which checks not only whether automatic analysis retrieves a plausible analysis, but also whether the alternative interpretation is manifestly implausible. Essentially, all of the example sentences above would be rejected, and only sentences in which both semantic and morphological evidence indicate one particular structure would be considered. The alternative would be a *bootstrapping* approach: simply parse great amounts of text automatically, and assume that the correct associations will have been retrieved more often than not. The net effect would then still allow us to learn the correct rather than the incorrect preference, even though some of the evidence is incorrect. A general parser that maintains an accuracy of over 90% even for labelled dependency edges would be a key component for such a setup.

Bibliography

Steven Abney. Parsing By Chunks. In Carol Tenny, editor, *The MIT Parsing Volume, 1988–89*. Center for Cognitive Science, MIT, 1989.
`www.vinartus.net/spa/89d.pdf`

> Abney introduces the notion of *chunks* as sequences of a content word and its associated function words. Their existence is motivated both psychologically and computationally. In particular, context-free techniques can be used for half of the parsing problem, and more complicated techniques such as subcategorization calculations need only be applied on the level of chunks. A toy grammar and a two-stage LR parser is presented that analyses English in this way.

Steven Abney. Statistical Methods and Linguistics. In Judith Klavans and Philip Resnik, editors, *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, pp. 1–26. The MIT Press, Cambridge, Massachusetts, 1996.
`citeseer.ist.psu.edu/abney96statistical.html`

> Abney argues that statistical methods for NLP are not imperfect approximations that can only be stopgap solutions, but are intrinsically necessary for the task. He cites aspects such as productiveness, gradual acquisition, disambiguation, gradedness of acceptability, error tolerance, and on-the-fly learning, which can only be explained by assuming weighted rather than binary grammars. A lucid final section anticipates the common objection 'But didn't Chomsky debunk all this ages ago?'.

Steven Abney, et al. Procedure for quantitatively comparing the syntactic coverage of English grammars. In E. Black, editor, *Proceedings of a workshop on Speech and natural language*, pp. 306–311. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1991. ISBN 1-55860-207-0.
`acl.ldc.upenn.edu/H/H91/H91-1060.pdf`

> This contribution proposes a solution for comparing the performance of different wide-coverage parsers on the same input, a problem that had not been adequately solved before. The widespread measures of recall and crossing brackets are introduced here for the first time. Although both the title and the text speak of comparing the coverage of different parsers, it is really concerned with computing accuracy.

Alfred V. Aho, et al. *Compilers: principles, techniques, and tools*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986. ISBN 0-201-10088-6.

This is the canonical reference for algorithms and data structures commonly employed in analysing non-natural languages.

Stefanie Albert, et al. TIGER Annotationsschema. Technical report, Universität Stuttgart, 2003.

These are the detailed annotation guidelines that were used for creation of the TIGER corpus. They differ from the guidelines for the older NEGRA corpus mostly by additional clarifications and not by explicit changes.

S. Aït-Mokhtar, et al. Robustness beyond shallowness: incremental deep parsing. *Natural Language Engineering*, 8(2/3):121–144, 2002.

The authors implement Abney's proposal of cascaded regular transducing for parsing french newspaper text. A precision/recall value of 92.6%/82.6% is reported for the task of subject detection.

Srinivas Bangalore and Aravind K. Joshi. Supertagging: an approach to almost parsing. *Computational Linguistics*, 25(2):237–265, 1999.

This Computational Linguistics article describes fundamentally the same system as the earlier one (Joshi and Bangalore, 1994). The dependency-sensitive approach was not pursued, because there just isn't a large enough LTAG parsed corpus and because this method is 'too much like full parsing'; but the other models were improved with larger corpora and smoothing, to 77% for unigram supertagging, and 92% for n-gram supertagging.

Srinivas Bangalore, et al. Heuristics and Parse Ranking . In *Proc. 4th International Workshop on Parsing Technologies (IWPT-95)*, pp. 224–233. Prague/Karlovy Vary, Czech Republic, 1995.
`arxiv.org/ps/cmp-lg/9508010`

This is a report on various filtering and ranking techniques in the XTAG system that were later superseded by supertagging. Three stages of heuristics are discussed: POS tagging (especially important for LTAG because of the much greater lexical ambiguity), tree filtering and weighting (e.g. discarding elementary trees that can be proven not to bee part of any complete tree), and global ranking (heuristics such as 'prefer arguments to adjuncts' or 'prefer low PP attachment'.) The heuristics themselves are handwritten and domain-independent, only their weight is affected by training.

Srinivas Bangalore, et al. An approach to Robust Partial Parsing and Evaluation Metrics. In *Proc. 8th European Summer School In Logic, Language and Information*. Prague, Czech Republic, 1996.
`citeseer.ist.psu.edu/srinivas96approach.html`

This paper reports on many aspects of the XTAG system, a 'wide-coverage grammar for English based on the FB-LTAG formalism'. The system architecture, performance on WSJ and IBM corpora, supertagging component and 'Lightweight Dependency Analyzer' are discussed. An evaluation method that

counts only the relations between correctly established chunks is proposed but not applied.

Michele Banko and Eric Brill. Scaling to Very Very Large Corpora for Natural Language Disambiguation. In *Meeting of the Association for Computational Linguistics*, pp. 26–33. 2001.
`citeseer.ist.psu.edu/banko01scaling.html`

Decrying the continuing use of established corpora despite the avalanche of new data, the authors train algorithms for confusion set disambiguation on an unprecedented 1,000,000,000 words, and find that none of them shows a saturation, i.e. the data is *still* sparse. For tasks where labelled training data are not free, various workarounds are suggested such as active learning and co-training.

Roberto Bartolini, et al. Hybrid Constraints for Robust Parsing: First Experiments and Evaluation. In *Proceedings of LREC 2004*, pp. 859–862. 2004.
`gandalf.aksis.uib.no/non/lrec2004/pdf/719.pdf`

The parser for Italian IDEAL+ is used to extract functional relations in the form of dependency edges. Both chunk parsing and partial dependency analysis are performed via finite-state automata and are completely unlexicalized; then constraints (subcategorization, ordering) are applied, and the system is evaluated on the task of subject/object disambiguation. The best configuration solves the task with a precision/recall of 92%/100%. The authors conclude that lexical information is essential to the task, provided that it is used probabilistically; for instance, verbs that have both transitive and intransitive readings should specify which variant is more probable.

G. Edward Barton. The Computational Difficulty of ID/LP Parsing. In *Meeting of the Association for Computational Linguistics*, pp. 76–81. 1985.
`citeseer.ist.psu.edu/bartonjr85computational.html`

Barton challenges Shieber's original assumption that ID/LP parsing is computationally feasible (such assumptions had previously been used to explain why natural language is 'efficiently parsable' for humans). It is shown that Shieber's algorithm, although better than the alternative of using the expanded CFG, is still exponential in the worst case, and that this is due to inherent difficulty rather than some specific weakness. In this situation, 'the linguistic merits of various theories are more important than complexity results.'

Roberto Basili, et al. Efficient Parsing for Information Extraction. In *Proc. ECAI 1998*, pp. 135–139. 1998.
`ai-nlp.info.uniroma2.it/zanzotto/1998_ECAI_BasiliPazienzaZanzotto.ps`

The CHAOS (Chunk-Oriented Analysis System) for Italian is described. The keys to robust and efficient behaviour are said to be stratification and lexicalization: a finite-state chunker recognizes NP and VP of Italian, and verb subcategorization frames help with early clause boundary detection. A special purpose parser with a discontinuous grammar is used to establish inter-chunk dependencies (icd). Precision and recall of icds only is measured and amounts

to 75.2%/72.1% on the easiest text type (scientific and technical papers). This is a significant improvement over the earlier SSA (shallow syntactic analyzer).

Anja Belz. Optimisation of corpus-derived probabilistic grammars. In *Proceedings of Corpus Linguistics 2001*, pp. 46–57. Lancaster University, UK, 2001.
`citeseer.ist.psu.edu/article/belz01optimisation.html`

Belz points out that probabilistic grammars actually extracted automatically from tree banks are not good enough to compete with the state of the art; all published figures were achieved after so much additional hand-tweaking that the grammar bears little resemblance to the extracted one. However, truly automatic grammars might be useful as a starting point for systematic tweaking. Therefore she proposes to add *local structural context* to the models, that is, to do systematically what everybody does anyway manually.

Ann Bies, et al. Bracketing Guidelines for Treebank II Style Penn Treebank Project. Technical report, University of Pennsylvania, 1995.
`ftp://ftp.cis.upenn.edu/pub/treebank/doc/manual/root.ps.gz`

This document contains the guidelines that were distributed to annotators in the Penn Treebank project. Ultimately, this manual is responsible for the widespread complaints about lacking internal structure in the standard WSJ corpus. To be fair, the authors do deal thoroughly with many subtle distinctions e.g. of noun phrases (as evinced by subtitle such as 'An alphabetized Bestiary of treatments of measure and quantifier phrases'); however, much of these subtler distinctions were not consistently followed by all annotators.

Philippe Blache. Combiner analyse superficielle et profonde : bilan et perspectives. In *Proc. Traitement Automatique des Langues Naturelles, TALN-2005*, pp. 93–102. Dourdan, France, 2005.
`taln.limsi.fr/site/talnRecital05/tome1/P10.pdf`

Blache points out that hybrid systems are often proposed in order to reduce the complexity of an NLP task, but computational complexity does not always correspond to linguistic complexity. Four different methods of combining deep and shallow analyzers are characterized, and an architecture based on the 'Pipeline' method is proposed.

Tonia Bleam, et al. A Lexicalized Tree Adjoining Grammar for English. Technical report, University of Pennsylvania, 2001.
`ftp://ftp.cis.upenn.edu/pub/xtag/release-2.24.2001/tech-report.ps`

This manual describes the lexicalized Tree Adjoining Grammar of English developed at the University of Pennsylvania. It explains in detail how inversions, clefts, raising, extrapositions and many more phenomena were implemented using the XTAG development tools. The grammar has a home page at `www.cis.upenn.edu/~xtag/`.

Rens Bod and Ronald Kaplan. A probabilistic corpus-driven model for lexical-functional analysis. In *Proceedings of the 17th international conference on Computational linguistics*, pp. 145–151. Association for Computational Linguistics, Morristown, NJ, USA, 1998.

LFG-DOP is introduced as a combination of both grammar formalisms: the representation defined by LFG is used, while composition and decomposition operators are taken over from DOP. According to the authors, this combines the linguistic adequacy of LFG with the robustness of DOP.

Pierre Boullier. Supertagging: A Non-Statistical Parsing-Based Approach. In *Proc. 4th International Workshop on Parsing Technologies, IWPT-2003*, pp. 43–54. 2003.

`hmi.ewi.utwente.nl/sigparse/iwpt2003/boullier2.pdf`

At IWPT 2000, Martin Kay challenged the parsing community to find a way how to extract a faster, more permissive parser P' from a given statistical parser P so that P' can be practically used as a guide for P, for an overall speedup. Boullier presents a method that works for Earley parsing, which employs a finite-state model extracted from P that accepts a strict superset of L(P) as a guide for the actual Earley parsing. Compared with the normal predictor phase of the Earley parser, this model is nine times more accurate.

Sabine Brants, et al. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*. Sozopol, 2002.

`www.ims.uni-stuttgart.de/projekte/TIGER/paper/treeling2002.ps.gz`

The TIGER corpus is currently the largest treebank of written German. It is essentially a continuation of the earlier NEGRA corpus with a larger number of sentences and several systematic extensions to the annotation scheme (for instance, the arguments of complicated coordinations are now associated by extrasyntactic relations, a distinction is made between prepositional adjuncts and complements, etc.). The methods of corpus production are described (cascaded finite-state parsing, LFG, manual annotation) as well as the associated tools for tagging, corpus annotating, viewing, and searching.

Thorsten Brants. Inter-Annotator Agreement for a German Newspaper Corpus. In *Proc. 2nd International Conference on Language Resources and Engineering (LREC 2000)*, pp. 1435–1439. Athen, 2000a.

`www.coli.uni-sb.de/~thorsten/publications/Brants-LREC00.ps.gz`

The degree of inter-annotator agreement during the creation of the NEGRA corpus is measured, and categories that are particularly susceptible to annotator error are identified. Both for POS tagging and syntactic structure, rare categories are analysed much less consistently then frequent ones, but in absolute numbers disagreements about frequent categories contribute much more to the total error count.

Thorsten Brants. TnT – A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*. Seattle, WA, USA, 2000b.

This is the user manual for the distributed version of TnT. It stresses efficiency and applicability to any tag set as the main advantages. TnT is shipped with pre-compiled parameter files for English and German so that it can immediately be used.

Thorsten Brants, et al. Das NEGRA-Annotationsschema. Negra project report, Universität des Saarlandes, Saarbrücken, 1997a.
`www.coli.uni-sb.de/sfb378/negra-corpus/negra-corpus.html`

The guidelines for annotating the NEGRA corpus have since been superseded and evolved into the guidelines for annotating the newer TIGER corpus, hence this technical report is no longer available.

Thorsten Brants, et al. Tagging Grammatical Functions. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*. Providence, RI, USA, 1997b.
`acl.ldc.upenn.edu/W/W97/W97-0307.pdf`

The methods used in creating the NEGRA tree bank are discussed. Machine annotation is claimed to be both faster and better for frequently occurring phenomena, while manual annotation is required only for infrequently occurring constructions. Consequently, the annotator tool integrates statistical predictions of word and phrase categories and grammatical functions that the annotator must confirm of correct. An overview of the most frequent errors in the automatic prediction is given.

Christian Braun. Parsing German Text for Syntactico-Semantic Structures. In *Prospects and Advances in the Syntax/Semantics Interface, Proc. of the Lorraine-Saarland Workshop*. Nancy, 2003.
`www.loria.fr/~duchier/Lorraine-Saarland/braun.ps`

A system for creating a shallow semantic representation of German texts in the form of *partially resolved dependency structures* (PReDS) is described. Tokenization, topological parsing, chunking and PReDS building are pipelined according to the easy-first philosophy.

E. Brill and P. Resnik. A rule-based approach to prepositional phrase attachment disambiguation. In *15th International Conference on Computational Linguistics (COLING94)*. Kyoto, Japan, 1994.
`citeseer.ist.psu.edu/brill94rulebased.html`

Eric Brill solves the problem of PP attachment by the same kind of transformation-based error-driven learning that he already applied to POS tagging. He achieves 80.8% accuracy with 471 transformations. When uses the noun classes from WordNet to create more general rules, 81.8% accuracy is achieved with only 266 rules. The pros and cons are much the same as for transformation-based POS tagging: huge learning time, and the method is supervised-only.

Eric Brill. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4):543–565, 1995.
`citeseer.ist.psu.edu/article/brill95transformationbased.html`

Brill proposes that automatically learnt transformational rules combine advantages of empirical and rule-based approaches to classification: they create

models that consist of readily interpretable specific rules, yet can be learnt fully automatically from annotated corpora. The method is shown to be strictly more powerful than statistical decision trees. An application for POS tagging the WSJ corpus yields an accuracy 97.2%, which improves on the previous results of a Markov-model based tagger.

Eric Brill and Jun Wu. Classifier Combination for Improved Lexical Disambiguation. In Christian Boitet and Pete Whitelock, editors, *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pp. 191–195. Morgan Kaufmann Publishers, San Francisco, California, 1998.
`citeseer.ist.psu.edu/brill98classifier.html`

Three different empirical POS taggers are applied to the WSJ corpus, and their high complementary error rate is taken as an indication that combining them could yield an accuracy higher than that of either alone. Altogether, a simple majority voting scheme manages to reduce the overall error rate from 3.2% to 3.0%, and using example-based learning for determining which tagger to believe in cases of conflict gives a slight further improvement to 2.8%.

T. Briscoe and J. Carroll. Apportioning Development Effort in a Probabilistic LR Parsing System Through Evaluation. In *Proceedings Sixth Workshop on Very Large Corpora*, pp. 92–100. 1996.
`www.aclweb.org/anthology/W96-0209`

This paper describes ongoing work on an English wide-coverage parser that uses both a unification grammar and a variant probabilistic LR PCFG parser. 79% coverage and 74%/73% of precision and recall are reported for the Suzanne corpus.

T. Briscoe, et al. Parser evaluation: A survey and a new proposal. In *Proceedings First Conference on Linguistic Resources*, pp. 447–455. Granada, Spain, 1998.

The authors systematically describe previously proposed methods of parser evaluation, and find them all lacking. A new complex evaluation scheme is proposed that counts the number of grammatical relations that a parser manages to establish correctly. This is primarily intended to be used in inter-system comparisons.

Ted Briscoe and John Carroll. Developing and Evaluating a Probabilistic LR Parser of Part-of-Speech and Punctuation Labels. In *Proc. 4th International Workshop on Parsing Technologies (IWPT-95)*, pp. 48–58. Prague/Karlovy Vary, Czech Republic, 1995.
`arxiv.org/abs/cmp-lg/9510005`

This is an earlier version of the authors' contribution to the 1996 SIGDAT workshop. In addition to the results published there, it describes an experiment where omitting punctuation marks (from those 106 sentences in the test set that contain any) lets recall decline by 10%.

Ted Briscoe, et al. Relational Evaluation Schemes. In *Proc. Workshop 'Beyond PARSEVAL'*, pp. 4–8. Las Palmas, Gran Canaria, 2002.
`www.cogs.susx.ac.uk/lab/nlp/carroll/papers/beyond02.pdf`

> An evaluation method for parsers is proposed that uses only grammatical relations between words. Extra slots in named relations can be used, e.g. to distinguish surface from logical subjects; also, a conjunction relation can relate three words rather than two. Using a separate (possibly underspecified) semantic representation to represent divergence between logical and surface form is also mentioned but not used. A system for extracting grammatical relations from existing treebanks through handwritten rules is described that recovers 'ncsubj' and 'dobj' relations from the BLLIP corpus at 84%/86%.

Chris Callison-Burch and Miles Osborne. Statistical Natural Language Processing. In Ali Farghaly, editor, *Handbook for Language Engineers*, number 164 in CSLI Lecture Notes, chapter 7. CSLI Publications, 2003.

> This is the followup contribution to Megerdoomian (2003); rather than with the corpora themselves it deals with the algorithms used for exploiting them. Modelling, parameterizing and estimating are discussed, different types of probability models are contrasted, and evaluation and error analysis are motivated.

J. Chanod and P. Tapanainen. Statistical and constraint-based taggers of French. Technical report, XEROX, 1994.
`citeseer.ist.psu.edu/chanod94statistical.html`

> The authors set out to refute earlier claims that statistical methods are superior to linguistic rules for POS tagging. They retrain the XEROX tagger on a complex tag set for French and then spend the same amount of time in writing a constraint-based tagger manually, which makes fewer than half the number of errors.

Jean-Pierre Chanod and Pasi Tapanainen. Creating a tagset, lexicon and guesser for a French tagger. In *ACL SIGDAT workshop:From Texts To Tags: Issues In Multilingual Language Analysis*, pp. 58–64. University College Dublin, Ireland, 1995.

> This is a closer description of components used in the experiments published in the previous contribution. The authors motivate why they mostly remove gender, person, tense and mood information from the original tag set, but introduce some other distinctions. A disambiguation component based on productive endings is also described that classifies 85% of unknown tokens correctly (and over-generates for 7% more).

Eugene Charniak. A Maximum-Entropy-Inspired Parser. In *Proc. 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 132–139. Seattle, Washington, 2000.
`citeseer.ist.psu.edu/charniak99maximumentropyinspired.html`

> Charniak presents a variation on Collins's WSJ parser that increases precision and recall to 90.1% on sentences of up to 41 words. Instead of a treebank grammar, a third-order Markov grammar is used; the many parameters needed

for that model are estimated by maximum entropy methods. Charniak claims that this result disproves that the limit to statistical treebank parsing has been reached.

Eugene Charniak and Mark Johnson. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *Proc. ACL 2005*, pp. 173–180. Association for Computational Linguistics, Ann Arbor, Michigan, 2005.
`http://www.aclweb.org/anthology/P/P05/P05-1022`

A PCFG in combination with a maximum entropy reranker is described that achieves a labelled bracket f-score of 91.0% on the WSJ corpus. The Charniak parser is used, but because 50-best parsing would be prohibitively expensive, a coarser version of the model is run first to produce a packed parse forest, which is then evaluated with the full model. With this technique, 50-best parsing takes not more than three times as much time as 1-best parsing. The fine-grained model itself achieves only an f-score of 89.7%. The reranker now selects one of the 50 best parses instead of the best, depending on a multitude of tree features such as 'are the two conjuncts of a coordination structurally identical?', 'are they of the same length?', 'is this a right-branching structure?', 'how many preterminals does this node dominate and what category is it?', 'what POS categories appear to its left and right?', 'what node categories appear to its left and right?', etc. A maximum entropy estimator is used to compute weights for each of the 1,148,697 features (that appear at least 5 times in training). This selection leads to the new record f-score of 91.0% (where a perfect reranker would achieve 96.8%).

Eugene Charniak, et al. Taggers for Parsers. *Artificial Intelligence*, 85:45–57, 1996.

The effects of different varieties of statistical POS taggers with a PCFG chart parser are compared. In particular, a standard Markov-mode based tagger is compared with a model that can return more than one tag for a word, and is found to yield almost no improvement. The conclusion is drawn that for this kind of grammar (a PCFG on category symbols), multi-tagging is not necessary.

Noam Chomsky. *Syntactic Structures*. 's-Gravenhage, Mouton, 1957.

This work is important because of its seminal character rather than for the exact formalism it proposes. While many details and subsystems of transformational grammar are rather different today from what was proposed in 1957, this book was the first record of the arguments used to show that natural language can be viewed as a kind of formal language, and what particular kind of formal language must be assumed. Some common misunderstandings of the arguments in this book are addressed by Abney (1996).

Kenneth Church. On Parsing Strategies and Closure. In *Proc. 18th Meeting of the ACL*, pp. 107–111. Association for Computational Linguistics, Morristown, NJ, USA, 1980.
`portal.acm.org/citation.cfm?id=981469`

Church proposes YAP, an improvement on Marcus's PARSIFAL that can analyse more complex, acceptable utterances with a finite stack and fail on unacceptable ones. He argues that human performance is actually easier to duplicate than competence and that finite-state methods with a finite stack are sufficient. Combined with Marcus's Determinism Hypothesis, they correctly disallow multiple center-embedding, but allow unbounded movement. YAP will have to have a garbage collector to clear the stack of useless nodes, and that means it must have a definite closure policy. Both previous strategies (Kimball's Early Closure and Frazier's Late Closure) are too extreme; Church proposes a compromise similar to late closure, but allowing early closure to clear the stack for right recursive rules; in contrast to Kimball's idea it waits for two nodes instead of one.

Kenneth Church. What's Happened Since the First SIGDAT Meeting? In *Proc. Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, p. 1. 1999.

`acl.ldc.upenn.edu/W/W99/W99-0601.pdf`

This is the foreword of the 1999 SIGDAT conference on very large corpora, which reflects on the very rapid change to the field since 1993.

Kenneth Church and Ramesh Patil. Coping with Syntactic Ambiguity or How to Put the Block in the Box on the Table. *Comput. Linguist.*, 8(3-4):139–149, 1982. ISSN 0891-2017.

Certain constructions of English such as prepositions, adjuncts, coordination exhibit an *all-way ambiguity*, that is, every possible binary tree is actually a valid parse. Methods are discussed to represent such very ambiguous attachments efficiently in an ATN grammar, so that the processor can delay its decision 'until it discovers some more useful constraints' (by this, they mean 'during later semantic processing').

Kenneth Ward Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the second conference on Applied natural language processing*, pp. 136–143. Association for Computational Linguistics, Morristown, NJ, USA, 1988.

`acl.ldc.upenn.edu/A/A88/A88-1019.pdf`

This paper describes an early POS tagger that learns lexical and trigram probabilities from the Brown corpus. Examples show how some of the symbolic lexical disambiguation rules in the older Fidditch parser can be reformulated as simple trigram probabilities; long-distance dependencies are concluded to be 'not very important, at least most of the time'. Church reports '95–99%' of accuracy.

Stephen Clark and James R. Curran. The Importance of Supertagging for Wide-Coverage CCG Parsing. In *Proceedings of Coling 2004*, pp. 282–288. COLING, Geneva, Switzerland, 2004.

`www.iccs.informatics.ed.ac.uk/~stephenc/papers/clark_coling04.pdf`

A maximum entropy supertagger is used to reduce the alternatives from which a full CCG parser has to select. In conjunction with various techniques for reducing the size of the search space, the training of the entire parser can now be done on a single computer (if it is a really, really big computer). The resulting system can parser 98.5% of the WSJ test set. The authors consider their system 'the fastest linguistically motivated parser in the world'.

Stephen Clark, et al. Building Deep Dependency Structures with a Wide-Coverage CCG Parser. In *Proc. 40th Annual Meeting of the ACL (ACL-2002)*, pp. 327–334. Association for Computational Linguistics, Morristown, NJ, USA, 2001.
`www.cs.toronto.edu/~gpenn/csc2501/clark_acl.pdf`

Combinatory Categorial Grammar is a mildly context-sensitive formalism in which categories express what other categories the word expects on its left and right, while typed combinatory rules are used to combine categories with each other. Dependency accuracy is evaluated on a custom-translated version of WSJ data, the *CCGbank*. A structural accuracy of over 90% is reported at a coverage of 'almost 98%'. Extracted objects are recovered correctly in 41.7% of all cases.

Michael Collins and James Brooks. Prepositional Attachment through a Backed-off Model. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pp. 27–38. Association for Computational Linguistics, Somerset, New Jersey, 1995.
`citeseer.ist.psu.edu/collins95prepositional.html`

Collins and Brooks solve the PP attachment problem by simple tuple-counting: when no instance of a 4-tuple is known from training, they back off to triples and pairs. The method achieves 84.5% accuracy on the same data as Ratnaparkhi and Roukos (1994). It is demonstrated that you should back off only to those tuples that contain the preposition. They also note that, contrary to popular wisdom, low-frequency tuples should *not* be discarded, since they contribute about 3% to the results.

Michael John Collins. *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania, 1999. Supervisor-Mitchell P. Marcus.
`people.csail.mit.edu/mcollins/papers/thesis.ps`

Collins implements a variant of PCFG where the rule probabilities are factored into three independent probabilities (head node category, left dependents and right dependents) and applies it to the standard WSJ corpus. Together with lexicalized probabilities of head/child pairs, special treatment for simple noun phrases and some other extensions this model ultimately achieves 88.0%/88.3% of precision and recall, which at that time was the best accuracy ever measured.

Michael A. Covington. A Dependency Parser for Variable-Word-Order Languages. Technical Report AI-1990-01, University of Georgia, Athens, Georgia 30602, 1990.

Covington argues at length that dependency relations are particularly well-suited for analysis of natural language, and gives an exhaustive left-to-right

algorithm for analysing language according to unrestricted dependency grammars. Although WCDG does not use this algorithm, it shares much of Covington's motivation and assumptions.

Berthold Crysmann, et al. An Integrated Architecture for Shallow and Deep Processing. In *Proceedings of ACL-2002, Association for Computational Linguistics 40th Anniversary Meeting, July 7-12*. Philadelphia, USA, 2002.
`www.dfki.de/dfkibib/publications/docs/wb-acl02.pdf`

This paper describes Saarbrücken's WHITEBOARD system for integrated deep and shallow processing. A multi-layered, object-oriented, partially XML-based chart structure connects, among others, their HPSG of German, the finite-state analyzer SPPC, Callmeier's PET, explanation-based NE recognition, GermaNet, and a stochastic topological parser. Altogether the coverage of the whole system was improved from 29% to 71% by the shallow techniques for the NEGRA corpus; no figures of parsing accuracy are given.

James R. Curran and Miles Osborne. A very very large corpus doesn't always yield reliable estimates. In *Proc. CoNLL-2002*, pp. 126–131. Taipei, Taiwan, 2002.
`www.cnts.ua.ac.be/conll2002/pdf/12631cur.pdf`

This is a response to Banko and Brill (2001), which advocated the use of 'very very large corpora'. 1,1 billion words of English were used in order to predict the frequencies of all words in the WSJ corpus. It is shown that the frequency of some words is still not predicted even remotely correctly, because words tend to occur in bursts rather than independently of each other.

Doug Cutting, et al. A practical part-of-speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pp. 133–140. Association for Computational Linguistics, Morristown, NJ, USA, 1992.
`acl.ldc.upenn.edu/A/A92/A92-1018.pdf`

A HMM POS tagger is presented which can be trained both supervised and unsupervised (i.e., with raw text and a lexicon). With unsupervised learning of 38 tags, 96% accuracy is achieved on the Brown corpus.

Michael Daum, et al. Constraint Based Integration of Deep and Shallow Parsing Techniques. In *Proc. 11th Conference of the EACL*, pp. 99–106. Budapest, Hungary, 2003.
`acl.ldc.upenn.edu/E/E03/E03-1052.pdf`

An earlier version of the grammar of German described here was tested on a corpus of online newscasts. POS tagging raised structural accuracy from 58.2% to 78.2%; additional chunk parsing raised the accuracy to 80.0%. This result first showed that statistical helper applications can function as an enabling technology that makes parsing of totally unrestricted text viable in the first place.

Michael Daum, et al. Automatic transformation of phrase treebanks to dependency trees. In *Proc. 4th Int. Conf. on Language Resources and Evaluation (LREC-2004)*, pp. 1149–1152. Lisbon, Portugal, 2004.
`nats-www.informatik.uni-hamburg.de/~wolfgang/papers/lrec2004.ps.gz`

The tool DEPSY is described, which can transform phrase structure annotations to dependency structure automatically. Lin's algorithm is extended with arbitrary callback functions that can be used to post-process the result of the head child algorithm. A plugin is described that translates the original NEGRA treebank to dependency structure as modelled by the dependency grammar of German described here with 99% structural accuracy.

Carl G. de Marcken. Parsing the LOB corpus. In *Proc. 28th Annual Meeting of the ACL*, pp. 243–251. 1990.
`acl.ldc.upenn.edu/P/P90/P90-1031.pdf`

Different variations of statistical POS taggers are tested on the Lancaster/Oslo-Bergen Corpus of English (50,000 sentences); a multi-tagging version of DeRoses's algorithm fares best, although I cannot understand the table that is given. A custom partial parser of English is also presented for parsing the LOB corpus, which contains quite ungrammatical sentences. It employs such diverse methods as 'CFG-like rules' to recognize 'lengthy, less structured constructions such as NPs', connection of neighbouring phrases, and 'an innovative system of deterministically discarding certain phrases, called lowering'. Parsing quality is demonstrated by selected examples, and the statement 'we are quite pleased with the results'.

T. Dean and M. Boddy. An analysis of time-dependent planning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pp. 49–54. 1988.

Time-dependent planning problems are defined as the task of reacting to predicted future events under varying amounts of available processing time. Anytime algorithms are characterized as algorithms that can be scheduled preemptively, return an answer even when interrupted, and improve over time in a well-defined manner.

Steven J. DeRose. Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics*, 14(1):31–39, 1988.
`acl.ldc.upenn.edu/J/J88/J88-1003.pdf`

The program VOLSUNGA described here was the first almost completely automatic POS tagger based on supervised learning. Building on the previous program CLAWS (Leech et al., 1983), DeRose redefines its optimality measure to arrive at the now standard trigram probability method.

Peter Dienes and Amit Dubey. Deep Syntactic Processing by Combining Shallow Methods. In *Proc. 41st Annual Meeting of the ACL*. Sapporo, Japan, 2003.
`www.coli.uni-sb.de/~dienes/dienes_dubey_acl03.ps.gz`

PCFG parsers must solve two tasks to deal with long-distance dependencies: finding the empty elements and connecting them to their antecedents. The authors show that if the empty element sites were known, the antecedents could be computed with an f-score of 91.4% on the WSJ corpus. A maximum-entropy-based tagger is presented that uses features such as POS tags and hand-written regular expressions as templates to predict location and type of the sites. It achieves a labeled f-score of 83.0% after some redefining of the task;

adding this as an oracle to the PCFG parser allows it to perform antecedent recovery with an f-score of 72.6% (as opposed to 49.7%).

Thomas G. Dietterich. Machine-Learning Research – Four Current Directions. *AI Magazine*, 18(4):97–136, 1997.

`www.cs.wisc.edu/~shavlik/Dietterich_AIMag18-04-010.pdf`

Dietterich describes ensembles of classifiers as one of four important trends in machine learning at that time. He notes that ensembles can be more successful than either of their members if the members have error rates below 0.5, and make somewhat uncorrelated errors. Different strategies such as subsampling, randomizing and input manipulation as well as combination strategies are presented. The question why ensembles are needed at all is answered with three points: because the training data might not define a single best classifier; to compensate for incomplete search algorithms; or to overcome representational inadequacies in the hypothesis space.

Stefanie Dipper, et al. DEREKO (DEutsches REferenzKOrpus). Technical report, Universität Stuttgart, Universität Tübingen, Stuttgart, Germany, 2002.

`www.sfs.nphil.uni-tuebingen.de/dereko/DEREKOReport.ps.gz`

The DEREKO project aimed to construct a very large partially parsed reference corpus of modern German (with a billion words). This report describes the sources, the types of annotated chunks and guidelines for using them, and the associated query tools.

Amit Dubey. What to Do When Lexicalization Fails: Parsing German with Suffix Analysis and Smoothing. In *Proceedings of ACL'05*, pp. 314–321. Association for Computational Linguistics, Ann Arbor, Michigan, 2005.

`www.aclweb.org/anthology/P/P05/P05-1039`

The NEGRA corpus is parsed with unlexicalized PCFG that are improved over earlier models in several respects. The class of unknown words is divided more finely by taking into account suffixes, adapting the algorithm used by Brants for TnT. The treebank is preemptively transformed to allow particular PCFG rules to be learnt as in Schiehlen (2004). Different variants of linear interpolation are used for smoothing (which is ordinarily not done on unlexicalized PCFG). The best of many configurations results in 76.3% labelled f-score (2-Markovised PCFG rules, four of the five transformations, Beam search and Brants smoothing.) (Note that the labels denote categories, not grammatical functions.)

Amit Dubey and Frank Keller. Probabilistic Parsing for German using Sister-Head Dependencies. In *Proc. 41st Annual Meeting of the Association of Computational Linguistics, ACL-2003*. Sapporo, Japan, 2003.

The Collins parser is retrained on the NEGRA corpus for parsing German, and its head-to-head dependency model proves to perform much worse there than in English. A modified sister-to-head model is proposed that improves realistic parsing performance from 68% to 71%.

Jason M. Eisner. Three New Probabilistic Models for Dependency Parsing: An Exploration. In *Proc. 16th International Conference on Computational Linguistics (COLING-96)*, pp. 340–345. Copenhagen, 1996.

`acl.ldc.upenn.edu/C/C96/C96-1058.pdf`

Eisner denounces probability models directly derived from parsers as non-intuitive and focusing on accidental details of syntax representation. He proposes 'bare-bones dependency structure' as a simple test-bed for parsing. A cubic-time bottom-up algorithm for computing the most probable dependency tree over a sentence is described, and three different probability models are investigated that can be used to guide it (called 'bigram lexical affinities', 'selectional preferences' and 'recursive generation'). A preliminary result of attaching 87% of words from the WSJ corpus correctly is announced.

J. D. Ferguson, editor. *Hidden Markov Models for Speech*. Princeton, New Jersey, 1980.

These are the proceedings of a seminar given about HMM at Princeton University in 1980. The whole empirical revolution in speech recognition has been described as indirectly originating from this event.

John Rupert Firth. *Studies In Linguistic Analysis*, chapter 1, pp. 1–32. Basil Blackwell, Oxford, 3rd edition, 1957.

Sisay Fissaha, et al. Experiments in German Treebank Parsing. In *Proc. 6th International Conference on Text, Speech and Dialogue (TSD-03)*. Ceske Budejivice, Czech Republic, 2003.

This is a report on the first corpus PCFG for German. An unlexicalized PCFG read off the NEGRA corpus without smoothing achieves a constituent f-score of 62% at a coverage of 98%. Parent encoding is used to raise the f-score above 70%, but at a coverage of only 70%. The authors observe the learning effects for different variants of the probability model and conclude that a larger training corpus would be helpful.

Kilian Foth. *Eine umfassende Constraint-Dependenz-Grammatik des Deutschen*. Universität Hamburg, 2004a.

`nats-www.informatik.uni-hamburg.de/pub/Papa/CdgManuals/deutsch.pdf`

This is the annotation guideline for the model of German discussed in this thesis. It describes the use of each of the used labels with numerous examples, and explicitly discusses how to resolve common cases where different alternatives of syntactic category, label, or structure seem possible. The manual is available as part of the CDG distribution.

Kilian Foth and Wolfgang Menzel. Subtree Parsing to Speed up Deep Analysis. In *Proc. 8th Int. Workshop on Parsing Technologies, IWPT-2003*, pp. 91–103. 2003.

`nats-www.informatik.uni-hamburg.de/~wolfgang/papers/iwpt2003.ps.gz`

An extension to the heuristic transformation algorithm is discussed in which subsequences of an utterance, delimited by various means, are first analysed in isolation and later recombined. This results in a major speedup for long utterances without any loss of parsing accuracy.

Kilian Foth, et al. *[X]CDG User Guide*. Natural Language Systems Group, Hamburg University, Hamburg, Germany, 2005a.
`nats-www.informatik.uni-hamburg.de/pub/Papa/CdgManuals/manual.pdf`

This is the user manual for the freely available WCDG implementation developed at Hamburg University. It describes both the command-line utility and the graphical browser and editor XCDG. The definition of the input language is fully specified, but the formalism itself is not discussed in detail.

Kilian A. Foth. *Disjunktive Lexikoninformation im eliminativen Parsing*. Studienarbeit, Universität Hamburg, Fachbereich Informatik, 1999a.
`nats-www.informatik.uni-hamburg.de/pub/Main/NatsPublications/foth-sa.ps.gz`

The data structures that represent dependency edges in the CDG program are extended so that they allow lexical ambiguity to be represented in a single data structure if it is *irrelevant* for the edge in question. Particularly if a grammar defines levels of analysis that ignore morpho-syntax, a massive reduction in space and time requirements is observed. An input notation for disjunctive lexicon items is also introduced; however, the mechanism is able to combine lexical variants whether they were originally notated as disjunctions or not.

Kilian A. Foth. *Transformationsbasiertes Constraint-Parsing*. Diplomarbeit, Universität Hamburg, Fachbereich Informatik, 1999b.
`nats-www.informatik.uni-hamburg.de/pub/Main/NatsPublications/foth-da.ps.gz`

This is the description of the design of the first transformation-based solution method implemented in WCDG, the error-driven method also employed for all experiments in this thesis. Note that this publication states that the algorithm cannot be applied to general word hypothesis graphs; this has since been fixed.

Kilian A. Foth. Writing Weighted Constraints for Large Dependency Grammars. In Geert-Jan M. Kruijff and Denys Duchier, editors, *COLING 2004 Recent Advances in Dependency Grammar*, pp. 25–32. COLING, Geneva, Switzerland, 2004b.
`nats-www.informatik.uni-hamburg.de/pub/Main/NatsPublications/radg2004.ps.gz`

This paper reports on some of the techniques and principles that were used to create the WCDG of German also used here. It briefly motivates grammar elements such as graded constraints, rich label sets, and multiple analysis levels. The treatment of phenomena such as coordination and valence and some general methods for tasks such as choosing weights or dealing with rule exceptions are presented.

Kilian A. Foth. Writing and Using CDG. Technical report, Universität Hamburg, Hamburg, Germany, to appear.

This is a general report on the possibilities of the CDG reference implementation maintained at Hamburg University. Because the grammar described in this thesis is so far the only large-scale grammar available for CDG, this manual aims to provide general advice for writing further grammars in the formalism.

Kilian A. Foth and Jochen Hagenström. Tagging for robust parsers. In *2nd Workshop on Robust Methods in Analysis of Natural Language Data, ROMAND2002*, pp. 21–32. Frascati, Italy, 2002.

`nats-www.informatik.uni-hamburg.de/pub/Main/NatsPublications/romand2002.ps.gz`

The first integration of POS tagging into WCDG is described and various experiments are conducted to measure its effects. For a realistic grammar of German, the accuracy can be improved by over 50%, while parsing time is decreased by 40%.

Kilian A. Foth, et al. The Utility of Supertag Information for Parsing German. 2005b.

An experiment is described where the WCDG of German described here is extended by a statistical supertagger that predicts subordination directions, labels, and dependent sets. Varying models of supertags are shown to increase the parsing accuracy significantly. In addition, an upper bound on the accuracy that can be achieved with perfect supertags is estimated.

Haim Gaifman. Dependency systems and phrase-structure systems. *Information and Control*, 8(3):304–337, 1965.

Gaifman shows that constituency and dependency grammar are strongly equivalent in the sense that they generate the same languages in equivalent ways under certain circumstances. This was sometimes taken to mean that there is no point in investigating any other formalism than constituency; however, the result actually only applies when comparing fully projective dependency grammar to constituency grammars in which every noun phrase is headed by a noun, etc.

Daniel Gildea. Corpus Variation and Parser Performance. In Lillian Lee and Donna Harman, editors, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pp. 167–202. 2001.

`www.icsi.berkeley.edu/~gildea/gildea-emnlp01.pdf`

Gildea reimplements Collins's treebank parser and finds that a simple version of it degrades from 86% to 80% when applied to the older Brown corpus. He concludes that the high accuracy variously reported for that treebank occurs partly because it is simpler and much more homogeneous than other corpora, and that particularly those features often added to PCFG-like parsers in order to improve performance (such as Collins's lexical bigrams) are of little use with respect to general applicability.

Jesús Giménez and Lluís Màrquez. Fast and Accurate Part-of-Speech Tagging: The SVM Approach Revisited. Technical report, TALP Research Center, Universitat Polytècnica de Catalunya, 2003.

`citeseer.ist.psu.edu/673021.html`

This is an application of the concept of a *support vector machine* to the task of POS tagging (including a short but very clear explanation of the concept). A seven-word window of words is exploited by features such as 'previous word is *the*' or 'next word may be noun'. A one-pass algorithm is proposed that

achieves 97% accuracy on WSJ data and has a number of other desirable properties: it is very efficient, training time is linear, and rich context features can be learnt with few parameters. The authors conclude that their program is suitable for real-life use.

Fred Glover. Tabu Search – Part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.

`leeds-faculty.colorado.edu/glover/TS - Part I-ORSA.pdf`

This is the first half of a report on the principles and applications of *tabu search* for combinatorial optimization. The central notion of the technique is to maintain a memory of forbidden search states in order to allow hill climbing searches to escape from local maxima.

However, not only this idea is introduced but a much more specific algorithm that determines many more details. For instance, Glover's version of tabu search explicitly takes the best step possible rather than just any step that is an improvement, and it complements tabu lists with *aspiration level functions* which allow the tabu status of a move to be overridden if its attains the aspiration level. Examples of application are given, and issues such as determining the optimal value of t (the number of steps for which a previous state remains tabu) are discussed.

Barbara B. Greene and Gerald M. Rubin. Automated grammatical tagging of English. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, 1971.

The 86 tags that were used in the Brown corpus are described in detail, as well as the program TAGGIT which was used for the initial annotation. It employed an exception dictionary, hundreds of suffix rules and thousands of context-frame rules which removed possible tags based on unambiguous words in the context; however the end result could still be ambiguous.

Barbara Grosz, et al. DIALOGIC: a core natural-language processing system. In *Proc, 9th conference on Computational linguistics (COLING-82)*, pp. 95–100. Academia Praha, , Czechoslovakia, 1982.

`portal.acm.org/citation.cfm?id=991828`

The DIALOGIC system is described. It was the core language analyzer of several information processing systems in the 1980s. It used a handwritten augmented phrase-structure grammar that combined attribute grammar with special machinery for semantic and pragmatic interpretation.

Jan Hajič. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In Eva Hajičová, editor, *Issues of Valency and Meaning. Studies in Honor of Jarmila Panevová*, pp. 12–19. Prague Karolinum, Charles University Press, 1998.

`ufal.mff.cuni.cz/pdt/Corpora/PDT_1.0/References/Czech_PDT.pdf`

The first two phases of development on the Prague Dependency Treebank are described. Morphology employs a thorough morphosyntactic tagset with over 3,000 items. The 'analytical' level decorates the surface string with a complete

labelled dependency tree (28 possible labels). The tectogrammatical level is based on 'Functional Generative Description as developed in Prague by Peter Sgall and his collaborators'. An overview of the editing tools used is given, and an application in statistical parsing is announced, which can already perform POS tagging with 90% accuracy.

Jan Hajič, et al. A Manual for Analytic Layer Tagging of the Prague Dependency Treebank. CDROM CAT: LDC2001T10., ISBN 1-58563-212-0, 2001. English translation of the original Czech version.
`shadow.ms.mff.cuni.cz/pdt/Corpora/PDT_1.0/References/aman_en.pdf`

This is the complete annotation specification for the Prague Dependency Treebank. The morphological, analytical and tectogrammatical level of annotation are described, including how to deal with ellipses, references, phraseologisms and many other special cases needed to deal with real newspaper text.

Keith Hall and Václav Novák. Corrective Modeling or Non-Projective Dependency Parsing. In *Proc. IWPT*. 2005.

Collins's and Charniak's parsers of Czech are combined with a maximum entropy model that tries to fix the regents of nonprojective dependencies. The parsers are trained by raising all edges as high as possible to make them projective. It turns out that almost all of the errors that such a parser makes because of the normalization are confusions between two words very close in the tree (1 or 2 links). The corrective model is trained on pairs of dependency trees and considers features such as form, lemma, tag, morphology etc. of the dependent, the suggested and the correct regent. Altogether the unlabelled dependency accuracy increases to 85% from 84.3%.

Mary P. Harper, et al. PARSEC: A Constraint-Based Framework for Spoken Language Understanding. In *Proc. International Conference on Spoken Language Processing*, pp. 249–252. Banff, Alberta, Canada, 1992.
`ftp://ftp.ecn.purdue.edu/speech//papers/icslp92_1.ps`

The system PARSEC is described, which implements and extends Maruyama's CDG in a quite different manner from WCDG. It is designed to be run on the output from actual speech recognition, which makes rather tight grammars and aggressive parallelization necessary to deal with the ambiguity in spoken language. Also, since only hard constraints are used the problem remains in $\mathcal{P}$, and consistency-based methods are sufficient to solve it. An implementation is announced which can actually run in $O(k + \log(n))$ time.

James Henderson. Neural Network Probability Estimation for Broad Coverage Parsing. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pp. 131–138. Association for Computational Linguistics, Morristown, NJ, USA, 2003. ISBN 1-333-56789-0.
`cui.unige.ch/~henderson/papers/henderson_eacl03.pdf`

A statistical left-corner parser is used to compute 88.8% of the structure of the standard WSJ treebank corpus. The new contribution is the nature of the

history-based probability model: where previous parsers used a finite handcrafted set of features to represent the unbounded parse history, Henderson uses Simple Synchrony Networks (neural nets whose hidden layer activations correspond to history feature vectors). These have the advantages that the set of features can be inferred from the training set, but it is still even possible to impose linguistically appropriate biases on the selection, such as structural locality.

Harald Hesse and Andreas Küstner. *Syntax der koordinativen Verknüpfung*. Number XXIV in studia grammatica. Akademie-Verlag Berlin, 1985.

The authors endeavor to give a complete description of the syntax of possible coordinations with the German conjunction *und*. They propose that the proper treatment of these possibilities in a dependency grammar of German requires a 'second-order syntax', in which sentences covered by the 'first-order' syntax (that is, well-formed sentences with no ellipsis or coordination phenomena) are systematically combined. Structures in this extended format can contain more than one regent for a word, and therefore are no longer strictly dependency trees.

Donald Hindle and Mats Rooth. Structural Ambiguity and Lexical Relations. In *Meeting of the Association for Computational Linguistics*, pp. 229–236. 1991. `citeseer.ist.psu.edu/hindle93structural.html`

This was the seminal paper about statistical PP attachment; it formulated the now canonical task: given a verb, a following noun and a preposition, decide whether the preposition attaches to the verb or the noun. The authors show that a complete world model is not necessary to make the majority of attachment decisions correctly. From the output from the Fidditch partial parser, 224,000 triples were extracted and used as training material. The Lexical Attraction score is computed as the logged ratio of the frequencies of verb/noun attachment for a particular triple. The method achieves 80% correctness, where human annotators achieve 85% (88% if they can see the entire example and not only the three head words). The authors conclude that a partially parsed corpus is a better source of information than a dictionary for this task.

Richard A. Hudson. Towards a computer testable word grammar of English. Number 2 in UCL Working Papers in Linguistics, pp. 321–339. University College London, 1989.

An algorithm for parsing in the formalism of Word Grammar is presented that is similar to Covington's, but requires newly added dependencies to fulfil a specific adjacency principle. Hudson has since abandoned this condition in favor of a 'surface structure principle', which says that the total dependency structure must *include* one that is equivalent to a phrase-structure analysis.

Richard A. Hudson. *English Word Grammar*. B. Blackwell, Oxford, UK, 1990. `citeseer.ist.psu.edu/richard91english.html`

Word grammar is based on direct relations between words rather than on phrases (except to model co-ordination). Although the focus in all publications so far has been on syntax structure, it is explicitly intended to be a

complete theory of natural language. This book treats both the formalism of word grammar in general and a partial model of modern English.

Mark Johnson, et al. Estimators for stochastic "Unification-Based" grammars. In *Proc. 37th annual meeting of the ACL*, pp. 535–541. Association for Computational Linguistics, Morristown, NJ, USA, 1999. ISBN 1-55860-609-3.
`portal.acm.org/citation.cfm?id=1034758`

Unification-based grammars are much harder to stochasticize than context-free methods, because maximizing the likelihood of a training corpus is infeasible for features derived from their context-sensitive constraints. A 'maximum pseudo-likelihood' estimator is proposed that optimizes the conditional likelihood instead. From 314 ambiguous Verbmobil sentences, it manages to pick the correct parse over 50% of times.

Aravind Joshi and Srinivas Bangalore. Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing. In *15th International Conference on Computational Linguistics (COLING94)*. Kyoto, Japan, 1994.
`acl.ldc.upenn.edu/C/C94/C94-1024.pdf`

In LTAG, the lexicon associates forms with elementary trees rather than lexemes, and this ambiguity is rather larger than for POS tagging. Supertagging is proposed quite analogously to POS tagging, and for the same reasons. When using the XTAG parsed WSJ corpus, a unigram model (on POS tags, not words) achieves only 15% supertag correctness, but 52% recall can be achieved by choosing the top 3 supertags for each word, and this already speeds up XTAG parsing by 87% 'whenever the supertagger succeeds'. With a more complicated greedy dependency-driven algorithm that can respond to interactions even beyond the n-gram window, 77% are achieved.

Narendra Jussien and Olivier Lhomme. Local search with constraint propagation and conflict-based heuristics. *Artif. Intell.*, 139(1):21–45, 2002. ISSN 0004-3702.
`portal.acm.org/citation.cfm?id=604205`

A hybrid algorithm for solving the generalized CSP is proposed: *decision-repair*. The main idea is to perform a local search (for its feasibility and robustness), but only on partial assignments, so that filtering can be used. It solves open-shop scheduling problems much better than before, finding new lower bounds on many known hard problems.

Fred Karlsson. Constraint grammar as a framework for parsing running text. In *Proceedings of the 13th conference on Computational linguistics*, pp. 168–173. Association for Computational Linguistics, Morristown, NJ, USA, 1990. ISBN 952-90-2028-7.
`acl.ldc.upenn.edu/C/C90/C90-3030.pdf`

Constraint grammar is presented as a formalism explicitly designed for parsing realistic text (as opposed to parsers implementing government & binding, LFG or GPSG). Constraints can be hard and soft, and their task is to discard as many of the alternatives for a word as possible. Tokenized, tagged, morphologically analysed English text is assumed. Syntax constraints assign

'flat, functional, surface labels': modifier and complement labels point in the correct direction but do not specify the head, e.g. 'DN>' means 'determiner for a noun to the right'. Thus, the output is always unique but underspecified.

Martin Kay. The Proper Place of Men and Machines in Language Translation. Technical Report CSL-80-11, Xerox Palo Alto Research Center, 1980.
portal.acm.org/citation.cfm?id=593157

Kay attacks the prevalent goals and methods of using computers for translation. Obviously, he says, we know so little about how language understanding works that doing it by computer is totally uncalled for: computers are good at automating what is well understood, but using them for something that we cannot sufficiently formalize is worse than useless.

Frank Keller. *Gradience in Grammar*. Ph.D. thesis, University of Edinburgh, 2000.

Keller demonstrates the pervasive nature of gradience as a grammatical phenomenon. He shows that different instances of gradience share some common properties such as ranking and cumulativeness. The distinction between soft and hard constraints is motivated, and optimality theory is extended to *linear optimality theory*, which is considered to 'contribute new insights to linguistic theory'.

C. T. Kitzmiller and Janusz S. Kowalik. Coupling Symbolic and Numerical Computing in Knowledge-Based Systems (Workshop Report). *AI Magazine*, 8(2):85–90, 1987.

In the 1980s, much thought was given back then was to integrating numeric optimization with expert systems, e.g. to help a user select the proper algorithm for his task, or to enable reasoning with ambiguous, contradictory and imprecise data. The report even claims that 'integrating formal mathematical methods and methods based on symbolic knowledge' could 'solve some of the problems currently deemed intractable'. 'Shallow' coupling is characterized as treating numeric routines as black boxes, i.e. the decision when to apply numeric methods and what to do with their results depends only on the variables problem's state variables. A 'deep' coupled system on the other hand knows the operating envelope of each component it has and selects them accordingly. Deeply coupled systems are said to increase robustness, performance, and maintainability.

Dan Klein and Christopher D. Manning. Accurate Unlexicalized Parsing. In *Proc. 41st Annual Meeting of the ACL (ACL-03)*, pp. 423–430. Association for Computational Linguistics, Morristown, NJ, USA, 2003.
portal.acm.org/citation.cfm?id=1075150

It is uncontroversial that unlexicalized PCFG are easier to create, smaller and faster, but lexicalized ones generally perform more accurately. The authors argue that the benefits of lexicalisation to parsing have been overestimated and that much of the benefit of lexicalization can be obtained by splitting states manually in linguistically motivated places. This allows an unlexicalized PCFG to achieve better results than even the early lexicalized ones.

Sheldon Klein and Robert F. Simmons. A Computational Approach to Grammatical Coding of English Words. *Journal of the Association for Computing Machinery*, 10:334–347, 1963.

`www.cs.wisc.edu/~sklein/Comp-Gram-Coder-JACM-1963.pdf`

This publication describes the very first automatic POS tagging program. It used relatively small dictionaries, suffix and special character analysis, and context frame tests. An accuracy of 90% is reported for a tag set with 30 categories.

Philipp Koehn and Kevin Knight. Empirical Methods for Compound Splitting. In *Proc. 11th Conference of the EACL*, pp. 187–193. Budapest, Hungary, 2003.

`acl.ldc.upenn.edu/E/E03/E03-1076.pdf`

Different strategies are proposed for splitting German compounds so that they can be processed by MT systems. A complex method that takes into account the frequencies of the word parts, their POS tags and evidence from a bilingual parallel corpus can process 99.1% of compounds correctly, but for actual MT it is outperformed by much simpler methods such as 'split into as many parts as possible'.

Nobo Komagata. Efficient Parsing for CCGs with Generalized Type-Raised Categories. In *Proc. International Workshop on Parsing Technologies (IWPT-97)*, pp. 135–146. Boston, MA, 1997.

`www.tcnj.edu/~komagata/pub/iwpt97.ps.gz`

Japanese regularly contains heavily elliptical coordinations such as 'John visited Mary and Ken [visited] Naomi', or even 'Diuretic is effective for persons with hypertension related to sodium, and beta blockers [is effective for persons with hypertension related] to the nervous system.' CCG is extended to Generalized Type-Raised CCG in order to model such phenomena. A parser for such a model is implemented and tested on 22 sentences from a Japanese medical textbook (but not for accuracy). The parser has a worst-case exponential running time; a more complicated worst-case polynomial algorithm is presented, but deemed superfluous because the observed running time of the parser is already cubic. This is in part attributed to the efficient representation of spurious (syntactic, but not semantic) ambiguity.

S. Kübler and E. Hinrichs. From chunks to function-argument structure: A similarity-based approach. In *Proceedings of ACL/EACL 2001*, pp. 338–345. Toulouse, France, 2001.

`citeseer.ist.psu.edu/ubler01from.html`

Although Abney's original chunk parsing paper speaks about a chunker and an attacher, so far almost all work has been on chunkers. This work presents the chunk parser TüSBL, which attempts to perform recombination as well as chunking. After the usual finite-state chunking, a memory-based learning method searches for complete matches with a pre-existing instance base, or, failing that, partial matches on the lexical level or even the POS level. The system is evaluated with respect to the functional labels that it assigns; for

German, 89.73% of all functional labels are are correct, but only 72% of all constituents are attached at all.

Sandra Kübler and Heike Telljohann. Towards a Dependency-Oriented Evaluation for Partial Parsing. In *Beyond PARSEVAL – Towards Improved Evaluation Measures for Parsing Systems (LREC Workshop)*. 2002.

`www.sfb441.uni-tuebingen.de/a1/Publikationen/dep_eval.pdf`

Evaluation of parsing the TüBA-D tree bank is discussed. This corpus consists of generalized constituent structures similar to NEGRA, with all heads explicitly annotated; this allows easy conversion to dependency structures as suggested by Lin (1995). Examples are given how a single attachment error would be punished only once by dependency recall rather than multiple times by constituent recall. The same holds for unattached phrases. The authors conclude that dependency evaluation is particularly suitable for partial parsers such as theirs.

Hagen Langer. *Parsing-Experimente – Praxisorientierte Untersuchungen zur automatischen Analyse des Deutschen*. Number 4 in Computer Studies in Language and Speech. Peter Lang, Europäischer Verlag der Wissenschaften, 2001.

The author reviews the current state of natural language parsing and presents his own system Gepard. Its grammar is a context-free production system that uses unification without recursive feature structures, but allows arbitrary extensions to be written directly in C. A forward-chaining Earley parser is used. A general grammar of German is presented that can analyse about one in three sentences from newspaper text (but accuracy is not reported). The bulk of the book is the detailed presentation of each of the thousand production rules; more than one constraint in the WCDG of German was directly inspired by these rules.

Lillian Lee. "I'm sorry Dave, I'm afraid I can't do that": Linguistics, Statistics, and Natural Language Processing circa 2001. In National Research Council, editor, *Computer Science: Reflections on the Field, Reflections from the Field*, pp. 111–118. The National Academies Press, 2004.

`www.cs.cornell.edu/home/llee/papers/cstb.pdf`

A lively recounting of the vicissitudes of statistical NLP throughout the 20th century: the postwar distributional research connected with Harris and Shannon, Chomsky's caustic critique of n-grams, and the resurgence of empiricism after its successes in speech recognition.

Geoffrey Leech, et al. The Automatic Grammatical Tagging of the LOB Corpus. *ICAME News*, 1(7):13–33, 1983.

The Lancaster/Oslo/Bergen corpus was automatically tagged with an extension of the earlier TAGGIT program. New features included manually assigned unigram penalties for rare readings, collocational probabilities, special rules for specific combinations of words and tags, an idiom detector, and finally a more complicated model to compute the most probable sequence, which can handle unknown word spans of unlimited length. The accuracy on the entire corpus is said to be 96%.

Roger Levy and Christopher D. Manning. Deep Dependencies from Context-Free Statistical Parsers: Correcting the Surface Dependency Approximation. In *ACL*, pp. 327–334. 2004.
`acl.ldc.upenn.edu/acl2004/main/pdf/270_pdf_2-col.pdf`

PCFG parsing is combined with loglinear postprocessing models that can reconstruct nonlocal dependencies which the constituent parser could not build. The three types of nonlocal dependencies in WSJ (null complementizers, dislocations, and control) are partially dependent on each other (e.g. dislocated elements can act as controllers as if they were still in their original place), so a three-step algorithm is developed that reconstructs one type at a time with separate feature sets and models trained for each step. The results are competitive by Johnson's PARSEVAL-based metric, but a better metric is also proposed which counts typed dependency relations instead. This is used to argue that context-free parsing is indeed less appropriate to German than to English.

Wolfgang Lezius, et al. A freely available morphological analyzer, disambiguator and context sensitive lemmatizer for German. In *Proceedings of the 17th international conference on Computational linguistics*, pp. 743–748. Association for Computational Linguistics, Morristown, NJ, USA, 1998.
`citeseer.ist.psu.edu/lezius98freely.html`

This paper describes a freely available (but closed) tagger that predicts morphology, category and lemma of German words. A large lexicon of base forms combined with algorithms for compounding, inflecting etc. provides morphological analyses. POS tagging is performed with Church's trigram algorithm and achieves 96% accuracy for the smaller of the two tag sets employed. Lemmatization in combination with POS tagging deduces base forms correctly 99.3% of the time.

Dekang Lin. A Dependency-based Method for Evaluating Broad-Coverage Parsers. In *IJCAI*, pp. 1420–1427. 1995.
`citeseer.ist.psu.edu/lin95dependencybased.html`

Lin proposes to evaluate parsers in dependency space even when they work in phrase space, and gives a general algorithm for producing dependency trees from phrase trees if necessary. In dependency space, precision and recall always coincide, since each word can only be attached correctly or not. Whereas in phrase structure space a wrong attachment can cause several constituents to be considered erroneous, it will always cause exactly one wrong dependency. This error measure can therefore be considered more intuitively adequate. Dependency representation also makes it easy to evaluate the performance of a parser selectively. If edge labels are used, common tasks such as judging the performance of a parser for subject detection are solved by counting only the corresponding edges.

Qing Ma, et al. Part of Speech Tagging with Mixed Approaches of Neural Networks and Transformation Rules. 1999.
`www2.nict.go.jp/jt/a132/member/murata/ps/nlprs99.ps.gz`

POS tagging of Thai suffers from very few available training data. A 3-layer feed-forward neural network with elastic input was previously used for tagging; it can use a variable-length context with an error back-propagation algorithm, making it suitable for this situation. Here it is complemented with a postprocessor that learns Brillian transformation rules from templates that are specially designed to cover conditions that the network is known to learn badly, e.g. those with logical OR relationships. This increases accuracy on ambiguous words by 1.1%; overall, 99.1% of words are tagged correctly.

David M. Magerman. Statistical decision-tree models for parsing. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pp. 276–283. Association for Computational Linguistics, Morristown, NJ, USA, 1995.
`portal.acm.org/citation.cfm?id=981695`

This is essentially a report on the application of SPATTER to the Wall Street Journal corpus, where it achieves labelled precision and recall of 84.5%/84.0%. The same radical conclusion is drawn as in the author's thesis: 'Syntactic natural language parsers have shown themselves to be inadequate' (because they can be replaced with induces decision trees).

David Mitchell Magerman. *Natural language parsing as statistical pattern recognition*. Ph.D. thesis, Stanford University, 1994.
`arxiv.org/abs/cmp-lg/9405009`

Magerman uses statistical decision trees for parsing of English. Since parsing amounts to a series of decisions, such as 'what category is this?' or 'where in the tree does this node go?', successive applications of classifiers can be used for making all the necessary decisions. Only binary statistical decision trees are used; they are grown automatically with a greedy maximum-likelihood algorithm and then pruned. The resulting SPATTER parser can disambiguate its test set correctly in 78% of all cases. Magerman triumphantly reports that a hand-written PCFG scores only 69% and concludes that he has proven that 'linguistics need not take part in the development of a parser'.

Mitchell P. Marcus. *Theory of Syntactic Recognition for Natural Languages*. MIT Press, Cambridge, MA, USA, 1980. ISBN 0262131498.
`portal.acm.org/citation.cfm?id=539702`

Hiroshi Maruyama. Structural disambiguation with constraint propagation. In *Proc. 28th Annual Meeting of the ACL (ACL-90)*, pp. 31–38. Pittsburgh, PA, 1990.
`portal.acm.org/citation.cfm?id=981828`

This paper defines the term *constraint dependency grammar*. Maruyama focuses on propagation algorithms and the ability of CDG to represent ambiguous structures efficiently as a network instead of enumerating alternatives. Also, the generative capability and the parsing complexity are discussed.

Benson Mates. *Stoic Logic*. University of California Press, 1961.

Ryan McDonald, et al. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proc. HLT-EMNLP*, pp. 523–530. 2005.
`www.seas.upenn.edu/~ryantm/papers/nonprojectiveHLT-EMNLP2005.pdf`

Unlabelled dependency trees are computed for sentences from the Prague Dependency Treebank with a Maximum Spanning Tree algorithm. In contrast to a previous method, an iterative algorithm is employed that can construct the non-projective trees needed for Czech natively. Different simplifications are necessary to circumvent the exponential number of margin constraints; 'Factored MIRA' works best and retrieves 84.4% of the regents in the standard PDT setup, against 83.3% before. The gain is greater if you count only the trees that are non-projective in the first place.

Mark McLauchlan. Thesauruses for Prepositional Phrase Attachment. In *Proceedings of CoNLL-2004*, pp. 73–80. Boston, MA, USA, 2004.
`www.cnts.ua.ac.be/conll2004/pdf/07380mcl.pdf`

McLauchlan reimplements the tuple-counting model of Collins and Brooks (1995) but with a different smoothing method. Two general-purpose thesauruses are used as sources of similarity measures. In addition, a specialized thesaurus was created by running the extraction algorithm on a corpus consisting solely of PP tuples. Using thesaurus smoothing gets the model from 84.3% to 85.2%; the general thesauruses actually worked better then the specialized one.

Karine Megerdoomian. Text Mining, Corpus Building, and Testing. In Ali Farghaly, editor, *Handbook for Language Engineers*, number 164 in CSLI Lecture Notes, chapter 6. CSLI Publications, 2003.

This is a general review of the motivation, methods, tasks and applications of corpus linguistics. Both the abstract issues (such as what a corpus is, and how to create one) and actual examples (what well-known corpora there are) are discussed. A final section explicitly juxtaposes the features of the symbolic and statistical paradigms.

Igor Mel'cuk. *Surface Syntax of English*. Benjamins, Amsterdam, 1987.

Mel'cuk demonstrates the second component (of six) of his Meaning-Text Theory, the interface between Surface Syntax and Deep Morphology, on the example of Modern English. Despite repeated warnings about the preliminary and incomplete nature of this exposition, it is in fact very thorough and detailed, containing complete lists of we at DAWAI would call relation types, ordering constraints, lexical attributes, and even lexicon items, and an annotator's guide with thousands of examples (and funny limericks). It is not an exaggeration to say that this book is a CDG of English waiting to be written down.

Wolfgang Menzel. Robust Processing of Natural Language. In *KI-95: Advances in Artificial Intelligence*, pp. 19–34. Springer-Verlag, Berlin, 1995.
`nats-www.informatik.uni-hamburg.de/~wolfgang/papers/ki95.ps.gz`

Robustness in language processing is defined as the consistency of system responses in the face of deviant input. Human hearers possess this ability even for input with multiple disturbances, while automatic systems are usually limited to anticipating specific errors or systematically retrying analysis with different rule sets when there is trouble. CDG is proposed as a formalism that combines integrated processing, where e.g. syntax and semantics can influence

each other in a helpful way, without the resulting decrease in robustness of tightly integrated approaches such as HPSG.

Wolfgang Menzel and Ingo Schröder. Decision procedures for dependency parsing using graded constraints. In Sylvain Kahane and Alain Polguère, editors, *Proc. Coling-ACL Workshop on Processing of Dependency-based Grammars*, pp. 78–87. Montreal, Canada, 1998.

`acl.ldc.upenn.edu/W/W98/W98-0509.pdf`

The WCDG formalism is introduced together with some possible solution methods, such as search, arc-consistency and pruning algorithms. The properties of robustness, uniform information representation and time-adaptive behaviour are explained.

Wolfgang Menzel and Ingo Schröder. Error Diagnosis for Language Learning Systems. *ReCALL*, special edition, May 1999:20 – 30, 1999.

`nats-www.informatik.uni-hamburg.de/~wolfgang/papers/recall99.ps.gz`

A diagnostic component using WCDG is described that is able to cope with syntactic as well as semantic irregularities by the same mechanism. Graded constraints provide the means both for robust parsing in the face of errors and for explicitly diagnosing them. A multi-level description is used that includes syntactic rules, general semantic rules such as sortal restrictions, and domain-specific knowledge about the objects in the toy world.

Wolfgang Minker. Comparative Evaluation Of a Stochastic Parser On Semantic And Syntactic-Semantic Labels. In *Proceedings of LREC 2004*. Lisbon, Portugal, 2004.

The ARISE European project develops train schedule inquiry answering systems. Arguments, dialog-related and task-related attributes are assigned to spans of spoken input by a stochastic semantic case grammar. This paper investigates the impact of also using SYLEX (syntactic analyzer of French) on the input. This is improves recall of semantic labels from 21% to 25%.

Marvin Minsky and Seymour Papert. *Perceptrons - Expanded Edition: An Introduction to Computational Geometry*. The MIT Press, 1969.

This book demonstrates some inherent limitations of artificial neural networks by describing tasks that they are fundamentally unable to solve. Some of the difficulties were later solved through the invention of hidden-layer neurons, but at the time of appearance the work almost singlehandedly stopped research in the field.

Steven Minton, et al. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. *Artif. Intell.*, 58(1-3):161–205, 1992. ISSN 0004-3702.

`www.isi.edu/sims/minton/papers/aij-mc.ps`

The authors reimplement a neural network that was successfully used to solve scheduling problems as a purely symbolic algorithm and find that its effectiveness is unaffected by the paradigm change. The key policy turns out to use complete assignments and transform them into better assignments, guided by

how many conflicts every alternative would remove. Conditions are investigated upon when the approach is likely to be most successful.

T. Mitamura and E. Nyberg. Controlled English for Knowledge-Based MT: Experience with the KANT System. 1995.

`citeseer.ist.psu.edu/mitamura95controlled.html`

The KANT system for authoring of Controlled English texts is described. Both vocabulary and syntax are heavily restricted to those constructions that are easy to process automatically. The authoring system checks its input automatically and suggests alternatives for problematic constructions; in some cases the author is asked for disambiguation decision that is then recorded by SGML in the source text. Altogether the number of analyses on their test set drops from 27 to 1.04; the authors declare that this increases not only the translation accuracy but also the clarity of source texts.

Gordon Moore. Cramming More Components Onto Integrated Circuits. *Electronics*, 19, 1965.

`www.cs.princeton.edu/courses/archive/fall05/frs119/papers/moore65.pdf`

This is the article that originally described the phenomenon later dubbed 'Moore's Law': the exponential growth rate observed for integration density on integrated circuits as a function of time. Later, the relation was often applied to other aspects of computers such as processor speed and bandwidth.

Alexis Nasr and Owen Rambow. SuperTagging and Full Parsing. In *Proceedings of the Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms*. 2004.

`www.cs.rutgers.edu/TAG+7/papers/nasr.pdf`

A chart parser of English for the TAG formalism is proposed that operates on supertags rather than words; when the correct supertags are known, an accuracy above 97% is measured, which is declared to be the degree to which supertags determine the entire parse. In contrast to the usual bilexical statistical parsers, this system uses no lexical information whatsoever in the parser proper, with the advantage that only the supertagger has to be retrained upon porting.

Joakim Nivre. An Efficient Algorithm for Projective Dependency Parsing. In *Proc. 4th International Workshop on Parsing Technologies, IWPT-2003*, pp. 149–160. 2003.

`hmi.ewi.utwente.nl/sigparse/iwpt2003/nivre.pdf`

A deterministic shift-reduce parser for dependency trees is proposed. The model only creates unlabelled, connected, projective syntax trees; this allows parsing in linear time. With the help of perfect POS information, 89% accuracy are achieved on a Swedish toy corpus. Since parsing decisions are never revoked, the choice of an arbitration policy (when to shift, reduce, or subordinate) is critical for the performance of this system. So far, very simple rules are employed such as 'subordinate the top word to the next word if their categories allow this'.

Joakim Nivre and Jens Nilsson. Pseudo-Projective Dependency Parsing. In *Proc. 43rd Annual Meeting of the ACL*, pp. 99–106. Association for Computational Linguistics, Ann Arbor, Michigan, 2005.
`www.msi.vxu.se/~nivre/papers/acl05.pdf`

Although dependency-based parsers can in principle represent discontinuous constituents directly, Nivre notes that most of them (notably his own memory-based shift-reduce parser) do not actually do so. For parsing the Prague Dependency Treebank, he proposes to modify the training set so that it is entirely projective, but to annotate the edges involved with labels which describe the normalizing process. After shift-reduce parsing, edges with such labels are lowered again. The best of the encodings given allows the correct reconstruction of over 99% of all nonprojective edges from these labels. Parsing the 7507 sentences in its test set yields attachment accuracies of 80.0%/72.7%, which is better than the 78.5%/71.3% obtained with no projectivising, and higher than previous nonprojective parsers on the same data (but still lower than Collins's and Charniak's projective parsers).

Joakim Nivre and Mario Scholz. Deterministic Dependency Parsing of English Text. In *Proceedings of Coling 2004*, pp. 64–70. COLING, Geneva, Switzerland, 2004.
`www.msi.vxu.se/~nivre/papers/coling04.pdf`

This paper presents the same empirical shift-reduce parser as the previous one, except that more than one lookahead token is considered now. When applied to an automatically transformed version of the WSJ corpus, an accuracy of 87.3% is achieved in linear parsing time. This figure is slightly below the best results for reconstructing WSJ trees; one reason for that is certainly the limited nature of the probabilistic model, but Nivre also quotes several other, such as the non-optimal POS tagger used and the lack of real dependency labels in the corpus.

Joakim Nivre, et al. Memory-Based Dependency Parsing. In *Proceedings of CoNLL-2004*, pp. 49–56. Boston, MA, USA, 2004.
`www.cnts.ua.ac.be/conll2004/pdf/04956niv.pdf`

This paper extends the previous shift-reduce parser from rule-based policies to an empirical one. Parse actions as well as dependency labels are predicted by memory-based learning (the ID1 algorithm as implemented in the TiMBL package). The flexibility of this algorithm when dealing with unknown configurations allows better results than a previous probabilistic model with a fixed backoff sequence. A hand-annotated corpus of 6316 Swedish sentences is used for the experiments. On the development set, 81.7% labelled and 85.7% unlabelled accuracy are achieved with the best settings of all parameters. A statistic is also given about the accuracy of individual dependency types.

Geoffrey Nunberg, et al. Idioms. *Language*, 70(3), 1994.
`lingo.stanford.edu/sag/papers/idioms.pdf`

K. Papineni, et al. BLEU: a method for automatic evaluation of machine translation. Technical Report Technical Report RC22176 (W0109-022), IBM Research

Division, Thomas J. Watson Research Center, 2001.
`www1.cs.columbia.edu/nlp/sgd/bleu.pdf`

This technical report introduces the BLEU metric for evaluating machine translation systems. Essentially it evaluates the number of n-grams that are common to a candidate translation and human reference translations, normalized for sentence length. The metric can be efficiently computed and requires only an initial human effort, since the reference translations can be reused during the development of MT systems. BLEU ratings are shown to have excellent correlation with human judgement.

Núria Gala Pavia and Salah Aït-Mokhtar. Lexicalising a robust parser grammar using the WWW. In *Proc. Conference on Corpus Linguistics*. Lancaster, UK, 2003.
`www.up.univ-mrs.fr/delic/perso/gala/publis/ait-gala-cl03.doc`

A handwritten robust parser of French (XIL) is extended with PP attachment resolution based on Altavista search results. Ambiguous triples are translated into queries of the form 'A B NEAR C', and the resulting documents are parsed. Those cases in which the parser suggests a PP attachments between these words are then taken as evidence. These counts are then used to form a preference lexicon which is used in a second run on the original sentence. This increases PP attachment precision from 71% to 83% (but decreases recall from 92% to 85%.)

A. Ratnaparkhi and S. Roukos. A Maximum Entropy Model for Prepositional Phrase Attachment. In *ARPA Workshop on Human Language Technology*. Morgan Kaufmann, 1994.
`citeseer.ist.psu.edu/ratnaparkhi94maximum.html`

Ratnaparkhi extends the older PP attachment problem from judging a triple of words to a quadruple, by also looking at the object of the preposition in question. The task is solved with a maximum entropy model, and 81.6% correctness are reported on the WSJ corpus.

Adwait Ratnaparkhi. A Maximum Entropy Part-Of-Speech Tagger. In *Proc. Conference on Empirical Methods in Natural Language Processing*. University of Pennsylvania, 1996.
`www.cis.upenn.edu/~adwait/papers/tagger.ps`

Rathnaparkhi employs an ME approach to tag the WSJ corpus. The words and tags of the preceding and following two tokens are available as features, and features that occur fewer than 10 times are ignored. The initial performance is 96.4%, 86.2% on unknown words. Since some words (about, that, more etc.) are particularly difficult, accounting for 0.2 to 0.3 percent of all errors, the model is refined so that features can ask about the identity of the word in question itself, and treat e.g. 'that' differently from other RB words. This largely fails because the annotation itself is too inconsistent to make such fine distinctions; when restricting the experiment to sentences created by the same annotator, an increase to 97% is possible.

Adwait Ratnaparkhi. Statistical Models for Unsupervised Prepositional Phrase Attachment. In *COLING-ACL*, pp. 1079–1085. 1998.
`citeseer.lcs.mit.edu/ratnaparkhi98statistical.html`

The PP attachment quadruple task is solved with purely unsupervised learning, by only considering unambiguous sentences in the training set. For instance, 'lawyers in other jurisdictions' is a noun attachment because there is no verb to the left of the preposition. The model achieves over 80% correctness on English and Spanish texts.

Jane J. Robinson. DIAGRAM: a grammar for dialogues. *Commun. ACM*, 25(1):27–47, 1982. ISSN 0001-0782.
`portal.acm.org/citation.cfm?id=358387`

DIAGRAM was the grammar used in the DIALOGIC system for English dialogue interpretation. It was a large augmented phrase structure grammar that could deal with many complex phenomena; for instance, it correctly produced four semantic interpretations for 'She was given more difficult books by her uncle', modelling the ambiguity in PP attachment and the lexical ambiguity of 'more'. Many examples of coverage and even complete rules are given.

Emmanuel Roche and Yves Schabes. Deterministic Part-Of-Speech Tagging with Finite-State Transducers. *Computational Linguistics*, 21(2):227–253, 1995. ISSN 0891-2017.
`portal.acm.org/citation.cfm?id=211200`

A POS tagger using Brill's transformation-based method is converted automatically to the equivalent finite-state machine, which allows transformational tagging (but not training) in linear time.

Stuart J. Russell and Shlomo Zilberstein. Composing real-time systems. In *Proceedings of the IJCAI-91*, pp. 212–217. Sydney, 1991.

This paper shows how to construct practical applications by combining anytime algorithms, as defined by Dean. A LISP dialect is introduced in which such algorithms can be directly combined, and a method of evaluating systems with different performance profiles is given.

Jean E. Sammet. The early history of COBOL. *SIGPLAN Not.*, 13(8):121–161, 1978. ISSN 0362-1340.
`portal.acm.org/citation.cfm?id=808378`

This paper discusses the early history of COBOL, starting with the May 1959 meeting in the Pentagon which established the Short Range Committee which defined the initial version of COBOL, and continuing through the creation of COBOL 61. The paper gives a detailed description of the committee activities leading to the publication of the first official version, namely COBOL 60. The major inputs to COBOL are discussed, and there is also a description of how and why some of the technical decisions in COBOL were made.

G. Sampson, et al. Project APRIL: a progress report. In *Proc. ACL 1988*, pp. 104–112. Buffalo, N.Y., 1988.
`acl.ldc.upenn.edu/P/P88/P88-1013.pdf`

This is a progress report about Sampson's APRIL (Annealing Parser for Realistic Input Language) system, of which Sampson (1986) was 'a crude pilot version'. 50,000 words of the LOB corpus are used as a training set. On 50 tagged LOB sentences, an accuracy of 75.3% is reported. Many features of the APRIL system, right down to the transformation-based search strategy, are very similar to the WCDG system, although there is no genetic link at all.

G.R. Sampson. A stochastic approach to parsing. In *11th International Conference on Computational Linguistics (COLING86)*. Bonn, Germany, 1986.
`acl.ldc.upenn.edu/C/C86/C86-1033.pdf`

After CLAWS proved a great success in language processing without grammatical knowledge, Sampson attempted to use a similar method for unlexicalized syntax parsing (although today we would call it a rather different method in a similar paradigm): simulated annealing. A simple unlexicalized head-daughter probability measure is used. A transformation step can detach any node, reattach it, and change the new parent's label. Temperature is interpreted as the standard deviation of a Gaussian distribution whose samples are added to the difference in likelihoods. An example of a correct parsing run is given and the assurance is made that while not all runs are so successful, none yield 'totally crazy trees'. Extensions to incremental parsing or deep-structure parsing are envisaged.

Christer Samuelsson and Atro Voutilainen. Comparing a Linguistic and a Stochastic Tagger. In *Proc. 35th Annual Meeting of the Association for Computational Linguistics*, pp. 246–253. ACL, Madrid, Spain, 1997.
`acl.ldc.upenn.edu/P/P97/P97-1032.pdf`

The authors refute various doubts that had been expressed about their previous results. They show that their tag set is not trivial, that their corpus annotation was not primed by what the experimenters expected, and that human annotators really can achieve near-perfect agreement. In a comparison experiment, their system consistently achieves much better results than a Markov trigram tagger trained on the Brown corpus.

Beatrice Santorini. Part-of-speech tagging guidelines for the Penn Treebank Project. Technical Report MS-CIS–90-47, Department of Computer and Information Science, University of Pennsylvania, 1990.
`ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz`

This document defines the set of 36 part-of-speech tags used in the Penn Treebank, which has become the standard tag set for English. It aims to make further morphological classification unnecessary; for instance singular and plural nouns receive different tags. A large number of doubtful cases are explicitly discussed.

Michael Schiehlen. Combining Deep and Shallow Approaches in Parsing German. In Erhard Hinrichs and Dan Roth, editors, *Proc. 41st Annual Meeting of the ACL (ACL-03)*, pp. 112–119. 2003.
`www.aclweb.org/anthology/P03-1015.pdf`

The NEGRA corpus is analysed with different parsing algorithms in isolation and combined, and the results are evaluated using a dependency representation that allows for underspecification of very ambiguous phenomena such as prepositions. First, C4.5 machine learning is employed on the covered-nth-tag representation, leading to 77.2%/72.6% when all attachments are resolved; Schiehlen's own cascaded finite-state parser achieves 85.9%/76.1%. Many different combination schemes are tested, and the best result is greedy f-value optimization with an f-score of 85.4%.

Michael Schiehlen. Annotation Strategies for Probabilistic Parsing in German. In *Proceedings of COLING 2004*, pp. 390–396. COLING, Geneva, Switzerland, 2004.
`www.ims.uni-stuttgart.de/pub/vmob/papers/Schiehlen:coling041.pdf`

An unlexicalized probabilistic parsing model for NEGRA is presented that uses various 'linguistically inspired' transformation and annotation schemes: transformations are applied to the corpus before training and later removed from the parser output before measuring accuracy. No scheme that reduces the number of category symbols has a positive effect, but many that increase the number of categories do; for instance, when the special German PP with 'von' that function as genitives are given a special category `PP-PG` instead of `PG`, accuracy improves.

Many of these transformations use information from NEGRA's edge labels to detect unusual constituents and mark them as special, effectively introducing lexicalisation at key points. Also, trees that are considered multiple sentences are split into smaller sentences, a NE recognizer is applied, etc. Altogether these extra measures raise the dependency f-score to from 78.1% to 81.2% on the test set.

Anne Schiller, et al. Guidelines für das Tagging deutscher Textcorpora. Technical report, Universität Stuttgart / Universität Tübingen, 1999.
`www.sfs.nphil.uni-tuebingen.de/Elwis/stts/stts-guide.ps.gz`

This text defines the set of 54 part-of-speech tags that is now generally used when performing automatic analysis of German. The tag set is structured hierarchically by major and minor categories, and optionally morpho-syntactic features. Detailed guidelines are specified for annotating new text with this tag set.

Helmut Schmid. Part-of-Speech Tagging with Neural Networks. In *Proc. International Conference on New Methods in Language Processing*. Manchester, UK, 1994a.
`acl.ldc.upenn.edu/C/C94/C94-1027.pdf`

The POS tagging problem is solved with a 2-layer perceptron network. Input features are the output prediction of the preceding and the lexical probabilities of the following words. On Penn Treebank data, the network performs better than the Xerox tagger and comparable to Cutting's HMM tagger, but copes better with a smaller training set (since it has to determine only 13,824 parameters).

Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proc. International Conference on Computational Linguistics, (COLING-94)*, pp. 172–176. Kyoto, Japan, 1994b.

`www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger1.ps.gz`

An n-gram POS tagger is presented which does not obtain its transition probabilities through maximum likelihood estimation, but from binary decision trees on features such as 'does the previous word have tag t?'. With the ID3 algorithm, contexts can be reliably pruned to those that provide a significant information gain, so that even a quadrogram version is possible. The resulting program achieves 96.32% accuracy on Penn Treebank data, more than a HMM tagger with the same data. Furthermore, the decision tree algorithm degrades more gracefully if the size of its training set is reduced.

Helmut Schmid. Improvements in Part-of-Speech Tagging with an Application to German. Technical report, Universität Stuttgart, Stuttgart, Germany, 1996.

`www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger2.ps.gz`

Improvements are proposed to a decision-tree POS tagger in order to make it work better on German, where morphology creates much more lexical parameters (to the point where 50% of all types are unknown words), but less training data is available. Measures such as equivalence classes for words with identical sets of possible tags, prefix lexica, and unsupervised learning of possible tag sets for unknown words lead to an accuracy of 97.53% in a preliminary experiment. Ideas for further improvements are discussed that would transcend the n-gram paradigm.

Ingo Schröder. *Analyse natürlicher Sprache durch Beschränkungserfüllung*. Master's thesis, Universität Hamburg, Fachbereich Informatik, 1995.

An early version of the CDG program is presented that was the first functional implementation of WCDG.

Ingo Schröder. *Integration statistischer Methoden in eliminative Verfahren zur Analyse natürlicher Sprache*. Diplomarbeit, Universität Hamburg, Fachbereich Informatik, 1996.

`nats-www.informatik.uni-hamburg.de/~ingo/papers/da.ps.gz`

The advantages of automatically acquired grammars over handwritten ones are motivated (effort, human error, generality etc.). WCDG is extended with an empirically computed valuation of individual attachments to guide the consistency-based solution methods. On a Verbmobil dialogue, the combined system achieves a better combination of disambiguation and correctness than the hand-written base grammar; however, the available training corpus proves too small to support reliable analysis with full disambiguation.

Ingo Schröder. *Natural Language Parsing with Graded Constraints*. Ph.D. thesis, Dept. of Computer Science, University of Hamburg, Germany, 2002.

`nats-www.informatik.uni-hamburg.de/~ingo/papers/thesis.pdf`

Schröder proposes the WCDG formalism for the treatment of the central phenomenon of *gradation* in natural language. Syntax and semantics of the WCDG

constraint language are reviewed, and the available solution algorithms are presented. Using a grammar for parsing Verbmobil appointment scheduling dialogues (an early version of the grammar discussed here), the capability for robustness against deviant input, error diagnosis, anytime behaviour, and external confidence assessments is demonstrated.

Ingo Schröder, et al. Learning Weights for a Natural Language Grammar Using Genetic Algorithms. In K. C. Giannakoglou, et al., editors, *Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems*, pp. 243–247. CIMNE, Barcelona, 2002.

`nats-www.informatik.uni-hamburg.de/~wolfgang/papers/eurogen.ps.gz`

The weights of a comparatively small WCDG of German are systematically evolved by measuring the performance of different weight vectors on the same corpus. Compared with hand-assigned weights, the same accuracy can be achieved with the automatically evolved weights, or a slight improvement when genetic techniques are combined with manual initialization.

Michael Schulz. *Parsen natürlicher Sprache mit gesteuerter lokaler Suche*. Diplomarbeit, Universität Hamburg, Fachbereich Informatik, 2000.

`nats-www.informatik.uni-hamburg.de/pub/Main/NatsPublications/SchulzDA.ps.gz`

A transformation-based solution method for WCDG is proposed that relies on *guided local search* rather than conflict analysis. Local optima are avoided not through explicit taboo lists, but by maintaining counters of features that have been observed in good and bad solution candidates, thus effectively changing the evaluation function used for hill climbing.

Lee Schwartz, et al. Disambiguation of English PP-Attachment using Multilingual Aligned Data. In *Proc. Ninth Machine Translation Summit*. New Orleans, LA, 2003.

`www.amtaweb.org/summit/MTSummit/FinalPapers/39-Aikawa-final.pdf`

A bilingual aligned English-Japanese corpus is used to improve PP attachment in machine translation. The key is that Japanese is syntactically unambiguous with respect to the classical quadruple task, so an aligned bilingual J-E corpus can be used as a corpus that is annotated as far as PP attachment is concerned. The MSR-MT system produces low PP attachments by default, so the corpus is used merely to collect information on when a verb has greater affinity for a PP than a noun does. With an additional dictionary of such V+P pairs, the resulting translations are judged by human informers to be significantly better, whether they were from English to Japanese or to Spanish.

Stuart M. Shieber. Sentence disambiguation by a shift-reduce parsing technique. In *Proceedings of the 21st annual meeting on Association for Computational Linguistics*, pp. 113–118. Association for Computational Linguistics, Morristown, NJ, USA, 1983.

`portal.acm.org/citation.cfm?id=981334`

A shift-reduce parser for English and various scheduling principles are described that model some preferences of human hearers in a natural way, such

as minimal attachment, right association, or lexical preference. The assignment of word category can be deferred in order to await better evidence, e.g. for an initial 'that', but since only a finite amount of information may be kept in a parsing state the parser still counts as deterministic.

Kiril Simov and Petya Osenova. A Hybrid Strategy for Regular Grammar Parsing. In *Proceedings of LREC 2004*, pp. 431–434. Lisbon, Portugal, 2004.
`citeseer.ist.psu.edu/639317.html`

BulTreeBank (`www.bultreebank.org`) is a corpus of analyses of Bulgarian sentences in HPSG format, but a partial, non-monotonic finite-state analyzer is used for bulk annotation. This paper describes the different regular grammars that are used to parse pieces of the input, and the concept of *dynamic networks of grammars* that allow their application in different orders. The current strategy includes easy-first, bottom-up treatment of base NPs, APs and verb nuclei, clitics, NEs, idioms, dates and multiwords; top-down clause and fixed-expression detection; and network-based treatment of PPs, infinitives, coordinations, relatives and discontinuities.

Wojciech Skut, et al. An Annotation Scheme for Free Word Order Languages. In *Proc. Fifth Conference on Applied Natural Language Processing (ANLP-97)*. Washington, DC, 1997.
`acl.ldc.upenn.edu/A/A97/A97-1014.pdf`

This is an early report on the principles and techniques used for creating the NEGRA corpus. Most important, nested phrase structure is rejected as suited only to configurational languages. Instead of trace/filler structures, non-contiguous constituents are allowed, while double syntactic roles are represented as secondary edges. A graphical annotation tool is described that integrates stochastic prediction components in order to ease the annotator's task.

Josep M. Sopena, et al. A Connectionist Approach to Prepositional Phrase Attachment for Real World Texts. In *Proceedings of the 36th annual meeting on Association for Computational Linguistics*, pp. 1233–1237. Association for Computational Linguistics, Morristown, NJ, USA, 1998.
`portal.acm.org/citation.cfm?id=980770`

The quadruple task is solved on examples from the WSJ corpus using a feed-forward neural network with one hidden layer. Wordnet is used but simplified to those classes that actually occur in the training and test sets. All classes of all senses of all words in the quadruple are presented as input simultaneously (and the output is a single bit, V or not V). 86.8% are achieved in the best configuration.

Jiri Stetina and Makoto Nagao. Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary. In *Proceedings Fifth Workshop on Very Large Corpora*, pp. 66–80. 1997.
`acl.ldc.upenn.edu/W/W97/W97-0109.pdf`

This work uses an ID3 decision tree to predict noun or verb attachment on standard data, and achieves an unsurpassed 88.1% of accuracy. The authors explain that they were inspired by the fact that Collins's backed-off model performs at 92.6% on instances when it does not have to back off (i.e. the entire quadruple has been seen before). Hence, one should try to increase the number of times that full matches happen. For instance, 'buy books for children' should be considered as the same as 'buy magazines for children'. To do this, the WordNet hierarchy was employed and a concept of semantic distance was introduced (a mixture of path length between two nodes and their absolute depths). An entire unsupervised algorithm for word sense disambiguation was invented to prepare the training data, since hand-disambiguating word senses in the WSJ corpus is infeasible. The resulting decision tree is extremely efficient and performs almost as well as human readers (when they see only the quadruple).

Jonathan Swift. Proposal for Correcting, Improving, and Ascertaining the English Tongue. Official petition, 1712.
`etext.library.adelaide.edu.au/s/s97p/`

A petition to the Earl of Oxford to establish a national academy for purifying and then preserving the English language, so that it might remain unchanged over time. Many arguments must appear bizarre to contemporary linguists, but in contrast to his more well-known pamphlet 'A Modest Proposal', this text is entirely serious.

Mary Swift, et al. Skeletons in the parser: Using a shallow parser to improve deep parsing. In *Proceedings of Coling 2004*, pp. 383–389. COLING, Geneva, Switzerland, 2004.
`www.cs.rochester.edu/ gildea/swift-coling04.pdf`

This work uses one parser (that of Collins) as an oracle for another (TRIPS). The domain that TRIPS parses is human/human dialogues in (simulated) emergency rescue situations, which is quite different from WSJ text, but it is reported that there are 'islands of stability' in Collins's output which are nevertheless useful. TRIPS itself is a GPSG/HPSG of English with 'fairly extensive coverage' operating 'close to real-time for short utterances' (1-9 words). The PCFG performance was only 32%/64% on the medical data, not surprising since it assumes systematically different phrase structure than TRIPS (see p. 386). Still, if you give the edges that it predicts a 3% bonus on their score in the TRIPS chart, it becomes 2.4 times faster, and sentence accuracy rises from 48.5% to 49.5%.

Pasi Tapanainen and Timo Järvinen. A non-projective dependency parser. In *Proc. 5th conference on Applied natural language processing*, pp. 64–71. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.
`portal.acm.org/citation.cfm?id=974568`

The older Constraint Grammar system for English (Karlsson, 1990) is extended to a full labelled dependency parser. Constraint Grammar output is underspecified in two ways: more than one tag may remain for a word, and the tags themselves may make partial predictions such as 'modifies a noun to

the somewhere to the right'. This work extends the system by adding explicit dependencies (which can even be nonprojective), although the output can still be underspecified. Rules can add or delete dependency links, but usually only add them. The actual tree is extracted from the intermediate result with the help of general heuristics. 95.3%/87.9% dependency attachment are claimed for newspaper text from the Bank of English corpus.

Kristina Toutanova, et al. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL*, pp. 252–259. 2003.
`www.stanford.edu/~krist/papers/tagging.pdf`

The authors motivate and present several additions to the previous existing statistical POS taggers of English, such as using both sides of context for each word, lexicalizing the context words in addition to the current word, and adding new features for classifying unknown words. The final result is an error rate reduction on the WSJ corpus of 4.4%. Even the authors admit that further advances in WSJ tagging is probably limited by treebank errors rather than algorithmic improvements.

Edward Tsang. *Foundations of Constraint Satisfaction*. Academic Press, London and San Diego, 1993.
`cswww.essex.ac.uk/CSP/edward/FCS.html`

This is a monograph about constraint satisfaction problems (the general problem of assigning consistent values to n-tuples). As well as the formal definitions and properties, it discusses search, consistency and repair algorithms. The last chapter also introduces soft constraints and defines both partial (PCSP) and optimization problems (CSOP).

Hans Uszkoreit. *Word Order and Constituent Structure in German*, volume 8 of *CSLI lecture notes*. University of Chicago Press, 1987.

Gertjan van Noord. Error Mining for Wide-Coverage Grammar Engineering. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pp. 446–453. Barcelona, Spain, 2004.
`odur.let.rug.nl/~vannoord/papers/error-mining.pdf`

A new technique for improving the Alpino parser of Dutch is described. Based on the notion of *parsability* of words and word sequences (the ratio of parsable to unparsable sentences from a large unannotated corpus in which it occurs), systematic errors in the parser can be found automatically. Both faulty or missing lexicon items and grammar rules can be detected automatically.

Based on these observations, improvements were made to the tokenizer, lexicon and grammar that increased the coverage from 91% to 95% of all sentences. The dependency accuracy rose from 84.5% to 86.5%. A new combination of suffix arrays and perfect finite hash automata was used to deal with the huge n-gram tables necessary.

C. J. van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.
`www.dcs.gla.ac.uk/Keith/Preface.html`

(The URL given above makes the full text available online.)

Martin Volk. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In *Proc. Corpus Linguistics 2001*. Lancaster University (UK), 2001.
`www.ifi.unizh.ch/cl/volk/papers/Lancaster_2001.pdf`

This was both the first application of PP attachment techniques to German, and the first to use search engine queries as its corpus. 3000 sentences from the *Computerwoche* CD were decorated with the usual quintuples as test data. AltaVista's `NEAR` operator was used to estimate co-occurrence frequencies in the entire German WWW. The work focuses on trading off reliability and coverage, and so not all cases are always solved. In the final configuration, lemmatization, compound analysis and backing off to one data point is used, and all remaining cases are simply assigned to `N`. This achieves 73.1% accuracy at full coverage.

Atro Voutilainen. A syntax-based part-of-speech analyser. In *Proceedings of the Seventh Conference of the European Chapter of the ACL*. Dublin, Ireland, 1995.
`acl.ldc.upenn.edu/E/E95/E95-1022.pdf`

Voutilainen challenges the dominance of data-driven methods for POS tagging. He notes that none of them seem to be able to break the '97% barrier' for English, while several hybrid systems achieve over 98%. The proposed EngCG is entirely rule-based: pattern-action rules on the context of words successively remove illegal tags until 93% to 97% of all words are totally disambiguated, while recall reaches 99.7%. With additional heuristic constraints, the remaining ambiguity can be halved yet again, with a recall of 99.5%.

Oliver Wauschkuhn. The influence of tagging on the results of partial parsing in german corpora. In *Proc. 4th International Workshop on Parsing Technologies*, pp. 260–270. Prague, Czech Republic, 1995.
`citeseer.ist.psu.edu/wauschkuhn95influence.html`

Wauschkuhn examines the influence of POS tagging on his shallow ChapLin parser. With perfect POS tags, 76% rather than 48% of sentences receive an unambiguous analysis. Statistical tagging makes 35% of sentences unparseable, but reduces the number of results. He concludes that tagging helps to reduce the number of results, at the cost of reduced coverage. No accuracy results are given, but with statistical tags, 79% of the unambiguous analyses are the same as without tags. Various hypothetical improvements to tagger integration are suggested.

R. Weischedel, et al. Coping with Ambiguity and Unknown Words through Probabilistic Models. *Computational Linguistics*, 19, 1993.

This is an early report on adding probabilistic knowledge to NLP systems in various ways. For POS tagging, PCFG parsing and semantic class extraction, the amount of training data necessary for an overall improvement is found to be quite small. For instance, 80 annotated sentences are enough to estimate PCFG rule probabilities which then reduce the ambiguity resolution error rate on sentences by 81%. From 20,000 words of MUC data annotated with semantic classes, probabilities for head+case frame+object triples can be estimated

which then predict 136 of 166 multi-way PP attachments correctly. For instance, the local modifier 'in Yunguyo' attaches correctly to 'exploded' rather than to 'dawn' or 'today'.

Greg Whittemore, et al. Empirical study of predictive powers of simple attachment schemes for post-modifier prepositional phrases. In *Proceedings of the 28th conference on Association for Computational Linguistics*, pp. 23–30. Association for Computational Linguistics, Morristown, NJ, USA, Pittsburgh, Pennsylvania, 1990.

`citeseer.ist.psu.edu/576083.html`

In an age when statistical methods were out of fashion, PP attachment was seen as 'essentially always predictable' by simple symbolic rules. In the very restricted domain of typed interactive communication in a travel planning system, a short algorithm combining Lexical Preference and Presupposition rules predicts almost all PP attachments correctly. The authors acknowledge that 'something about the typed interactive mode of communication' probably influences this predictability.

Hiroyasu Yamada and Yuji Matsumoto. Statistical Dependency Analysis with Support Vector Machines. In *Proc. 4th International Workshop on Parsing Technologies, IWPT-2003*, pp. 195–206. 2003.

`hmi.ewi.utwente.nl/sigparse/iwpt2003/yamada.pdf`

A deterministic shift-reduce dependency parser is described that achieves 90.0% accuracy on WSJ trees (converted to dependencies) with the best parameter settings. The statistical model is based on features such as 'POS tag' or 'POS tag of the leftmost child' of the left and right context of the current word. The probabilistic model is computed with the help of *support vector machines*, which use kernel functions to handle very large feature spaces efficiently and avoid overfitting to training data.

Anssi Yli-Jyrä. Coping with dependencies and word order or how to put Arthur's court into a castle. In Henrik Holmboe, editor, *Nordisk Sprogteknologi*, pp. 123–137. 2003.

`www.ling.helsinki.fi/~aylijyra/2004/NorfaYearbook2003.pdf`

This contribution develops an extended analogy about medieval England to illustrate what it actually means for a tree structure to be projective. It also extends the notion to non-crossing on more than one level, and so develops the *restricted multiplanarity hierarchy*.

Anssi Yli-Jyrä. Axiomatization of Restricted Non-Projective Dependency Trees through Finite-State Constraints that Analyse Crossing Bracketings. In Geert-Jan M. Kruijff and Denys Duchier, editors, *COLING 2004 Recent Advances in Dependency Grammar*, pp. 25–32. COLING, Geneva, Switzerland, 2004.

`www.ling.helsinki.fi/~aylijyra/dissertation/6.pdf`

A representation for dependency trees is defined axiomatically that can represent some non-projective dependency trees, namely those with a bounded *non-projectivity depth*. 'Colored' brackets are employed to guarantee that only

these can be represented. With such a model, finite-state methods can assign even non-projective structure to terminal strings, which is necessary for many natural languages such as Danish or Finnish.

Steve Young. Talking To Machines (Statistically Speaking). 2002.
`citeseer.ist.psu.edu/533053.html`

Spoken dialogue systems are proposed that use statistical techniques (Partially Observable Markov Decision Processes, Recursive Transition Networks, Stochastic Context-Free Grammars, Bayesian Networks, and n-gram models) for speech recognition, dialogue act detection, response generation and production. 'Minimal dependence on explicit rules' is stressed, and the trade-off between adequacy and demands on the training data is discussed.

Lotfi Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

Daniel Zeman. A Statistical Approach to Parsing of Czech. Technical report, Univerzita Karlova, Prague, Czech Republic, 1997.
`ufal.mff.cuni.cz/pdt/Corpora/PDT_1.0/References/1998-dan-pbml/pbml.pdf`

This is a summary of Zeman's dissertation on Czech dependency parsing. His extremely simple algorithm is: always add the edge to the incomplete dependency tree that has the highest unigram probability, and does not violate connectivity and projectivity. The unigram probability of an edge is just its relative frequency in the training data. 'None of the tested methods seems useful for parsing Czech with the data available.' The author attributes this to sparse data rather than the algorithm and lists seven inadequate simplifications made by the probability model so far.

Daniel Zeman and Zdenek Zabokrtsky. Improving Parsing Accuracy by Combining Diverse Dependency Parsers. In *Proc. IWPT*. 2005.
`http://ufal.mff.cuni.cz/czech-parsing/ZemanZabokrtskyIWPT2005.pdf`

Seven dependency parsers of Czech (Charniak, Collins, and five native Czech ones) are combined with different strategies to achieve better accuracy: Weighted voting gives the opinion of overall better systems more weight than others; Stacking attempts to learn which parser to trust for which decisions; Unbalanced combining just collects all edges proposed by at least half the parsers; Switching adds edges from different parsers as long as no cycle results. Weighted voting turns out to be best, raising structural accuracy to 87% from Charniak's 85%.