The German Statistical Grammar Model: Development, Training and Linguistic Exploitation

Sabine Schulte im Walde
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
schulte@ims.uni-stuttgart.de

December 2000

Contents

1 Introduction

Gramotron as a meta-project of the chair for theoretical computational linguistics defines a framework for developing and training statistical grammar models for the acquisition of lexicon information. The framework is language-independent and has been applied to German, English, Portuguese, and Chinese.

This report describes the development, the training and the exploitation of the German statistical grammar model.[1] I introduce into the necessary prerequisites for grammar development, i.e. a corpus as source for empirical input data (see Section 2), and a morphological analyser for analysing the corpus word-forms and assigning lemmas where appropriate (see Section 3). Section 4 gives insights into the development and structures of the German context-free grammar, which is followed by a description of the statistical training on the head-lexicalised probabilistic grammar variant in Section 5. Finally, Section 6 presents examples for the linguistic information within the statistic grammar model.

2 Corpus Preparation

As basis for empirical input data needed for the statistical training process, two sub-corpora from the 200 million token newspaper corpus *Huge German Corpus (HGC)* were created, (a) a sub-corpus `vfinal` containing verb-final clauses, and (b) a sub-corpus `relclause` containing relative clauses. Apart from non-finite clauses as verbal arguments, there are no further clausal embeddings, and the clauses do not contain any punctuation except for a terminal period. Table 1 summarises the aspects concerning the size of the corpora.

Corpus	Part	Tokens	Clauses	Token/Clause
HGC	`vfinal`	4,128,874	450,526	9.16
	`relclause`	10,137,703	1,112,010	9.12

Table 1: Size of sub-corpora

The reason for restricting the input data to verb-final and relative clauses was justified by the resulting simplified demands on the context-free grammar which was supposed to parse the input data: restricted sentence structures demanded only restricted grammar rules, thus the task of grammar development was simplified and shortened in time.

The corpus creation was based on methods and tools provided by the *IMS Corpus Workbench* developed by [Christ, 1994]. More specifically, the structural extraction was performed by the enclosed `Corpus Query Processor (CQP)` [Christ et al., 1999] and the queries

```
cqpcl 'HGC; [stts="$,"] [stts="KOUS"] [stts!="\$.*"]{0,25}
            [stts="V.*"] [word="[,.;:?\!]"]
            [word!="dass|ob" & stts!="PW(S|AT|AV)"];'
```

for verb-final clauses, and

```
cqpcl 'HGC; [stts="$,"] [stts="APPR"]? [stts="PREL(S|AT)"] [stts!="\$."]{0,25}
            [stts="V.*"] [word="[,.;:?\!]"]
            [word!="dass|ob" & stts!="PW(S|AT|AV)"];'
```

for relative clauses.

[1] Franz Beil started the work on the current version of the German grammar.

See some example sentences of verb-final clauses:

```
wie sich Hand oder Finger im künstlichen Raum auf die exakt gleiche Weise bewegen
dass Frauen die Wäsche ihrer Männer kaufen
bevor der Abstieg den sympathischen Familienbetrieb aus dem Sportforum ereilte
wenn ich diese Einrichtungen erst einmal erhalten kann
bis der Vorgänger sein Geschäft verrichtet hat
ob sie denn an diesem Tag mit etwas Besonderem rechneten
wie der Tag endet
dass ihm unmittelbar nach seiner Ankunft der Prozess gemacht wurde
ob wir Spiele spielen wollen
dass harte Kämpfe nötig sein werden
indem sie den anderen vom fahrenden Zug hinunterdrängten
```

and relative clauses:

```
das den Alterungsprozess der Fliegen aufhält
denen er jedoch aus Gleichgültigkeit nicht weiter nachging
der mit der Macht der Informatik einhergehen kann
in dem der Computer steht
was man gewöhnlich als Katalyse bezeichnet
welche die traditionellen Verhaltensweisen verändert
mit dem die Menschen den unaufhörlichen Strom neuer wissenschaftlicher Produkte betrachten
was euch am Leben erhält
die zuvor festgenommen worden waren
über deren Freilassung man sich aber bei Redaktionsschluss bereits geeinigt hatte
welche Rolle denn die Presse bei solchen Manövern spielt
```

3 Morphological Analyser

A finite-state morphology [Schiller and Stöckert, 1995] was utilised to assign multiple morphological features such as part-of-speech tag, case, gender and number to the corpus words, partly collapsed to reduce the number of analyses. For example, the word *Bleibe* (either the case ambiguous feminine singular noun 'residence', or a person and mode ambiguous finite singular present tense verb form of 'stay') is analysed as follows:

```
analyse> Bleibe
1. Bleibe+NN.Fem.Akk.Sg
2. Bleibe+NN.Fem.Dat.Sg
3. Bleibe+NN.Fem.Gen.Sg
4. Bleibe+NN.Fem.Nom.Sg
5. *bleiben+V.1.Sg.Pres.Ind
6. *bleiben+V.1.Sg.Pres.Konj
7. *bleiben+V.3.Sg.Pres.Konj
```

Reducing the ambiguous categories leaves the two morphological analyses

```
Bleibe { NN.Fem.Cas.Sg, VVFIN }
```

Apart from assigning morphological analyses the tool in addition serves as lemmatiser (cf. [Schulze, 1996]).

4 The Context-Free Grammar

The grammar is supposed to cover a sufficient part of the corpus, since in order to develop a statistical grammar model on basis of the grammar, a large amount of structural relations within parses is required. On the other hand, the grammar need not go into detailed structures for the relevant grammar aspects (such as subcategorisation frames) to be trained sufficiently, so the grammar structure is comparably rough.

The context-free grammar contains 5,033 rules with their heads marked. With very few exceptions (rules for coordination, S-rule), the rules do not have more than two daughters. The 220 terminal categories in the grammar correspond to the collapsed corpus tags assigned by the morphology. Grammar development is facilitated by (a) the grammar development environment of the feature-based grammar formalism YAP [Schmid, 1999], and (b) a chart browser that permits a quick and efficient discovery of grammar bugs [Carroll, 1997]. Figure 1 shows that the ambiguity in the chart is quite considerable even though grammar and corpus are restricted.

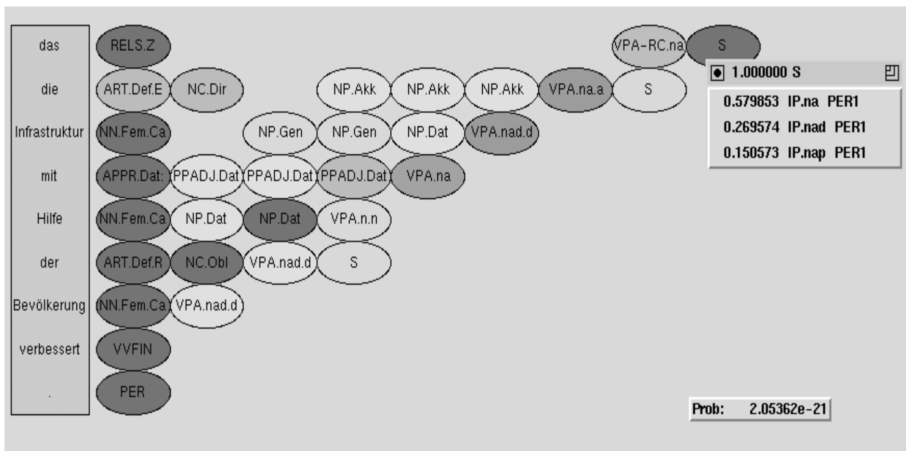


Figure 1: Chart browser for grammar development

The grammar covers 92.43% of the verb-final and 91.70% of the relative clauses, i.e. the respective part of the corpora are assigned parses.

The following sections 4.1 to 4.9 describe the German context-free grammar structures.

4.1 Clauses

The grammar expects verb-final or relative clauses for structuring the corpus data input described in Section 2, terminated by a terminal period. The top level rules are therefore

```
S -> IP.n'   PER1
S -> IP.na'  PER1
S -> IP.nd'  PER1
S -> IP.np'  PER1
S -> IP.nad' PER1
S -> IP.nap' PER1
S -> IP.ndp' PER1
S -> IP.ni'  PER1
S -> IP.di'  PER1
```

```
S -> IP.nai' PER1
S -> IP.ndi' PER1
S -> IP.nr'  PER1
S -> IP.nar' PER1
S -> IP.ndr' PER1
S -> IP.npr' PER1
S -> IP.nir' PER1
S -> IP.k'   PER1
```

with the IPs representing the underlying subcategorisation frames (cf. Section 4.2). The frame types can refer to active (`A`) and passive (`P`) clauses or copula constructions (`K`), as indicated on the next lower level, e.g.

```
IP.na -> IP-A.na'
IP.na -> IP-P.n'
IP.k  -> IP-K.n'
```

The connection between the different frame types will be explained in Section 4.2.

As mentioned above, clauses can be realised as verb-final clauses, exemplified by the transitive frame type `na`:

```
IP-A.na -> KOUS1 VPA.na.na'
```

or relative clauses:

```
IP-A.na -> VPA-RC.na.na'
```

with `VPX(-RC).<frame>.<frame>` indicating the maximal verb phrase level.

4.2 Verb Phrases

The grammar distinguishes four subcategorisation frame classes: active (`VPA`), passive (`VPP`), non-finite (`VPI`) frames, and copula constructions (`VPK`). A frame may have maximally three arguments. Possible arguments in the frames are nominative (n), dative (d) and accusative (a) NPs, reflexive pronouns (r), PPs (p), and non-finite VPs (i). The grammar does not distinguish plain non-finite VPs from *zu*-non-finite VPs. The grammar is designed to distinguish between PPs representing a verbal complement or adjunct: only complements are referred to by the frame type. The number and the types of frames in the different frame classes are given in Table 2, accompanied by example sentences.

Grammar rules concerning verb phrases satisfy their necessary complements by "collecting" them to the left: each finite verbal complex (cf. the description of verbal complexes following below) is automatically generated by all active, passive or infinitival phrases or copula constructions, depending on the mode of the verb complex. For example, the active present perfect verb complex *gestohlen hat* 'has stolen' is generated by all active verb phrases without any complements satisfied. Examples for the respective trees are in Figure 2.

German, being a language with comparatively free phrase order, allows for scrambling of arguments. Scrambling is reflected in the particular sequence in which the arguments of the verb frame are saturated. Compare Figure 3 as example of a canonical subject-object order within an active transitive frame and its scrambled object-subject order.

Abstracting from the active and passive realisation of an identical underlying deep-level syntax we generalise over the alternation by defining a top-level subcategorisation frame type, e.g. `IP.nad` for `VPA.nad`, `VPP.nd` and `VPP.ndp-s` (with `p-s` a prepositional phrase within passive frame types representing the deep-structure subject, realisable only by PPs headed by *von* or *durch* 'by'); see Figure 4 as example.

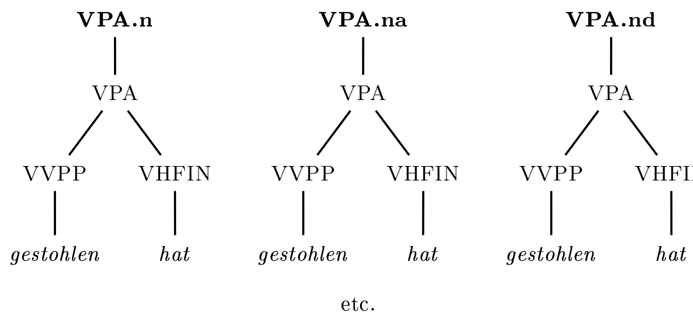


Figure 2: Verb frames without complements yet

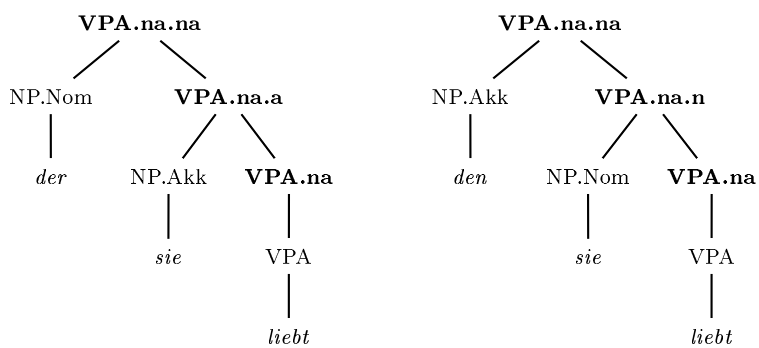


Figure 3: Realising scrambling effect in the grammar rules

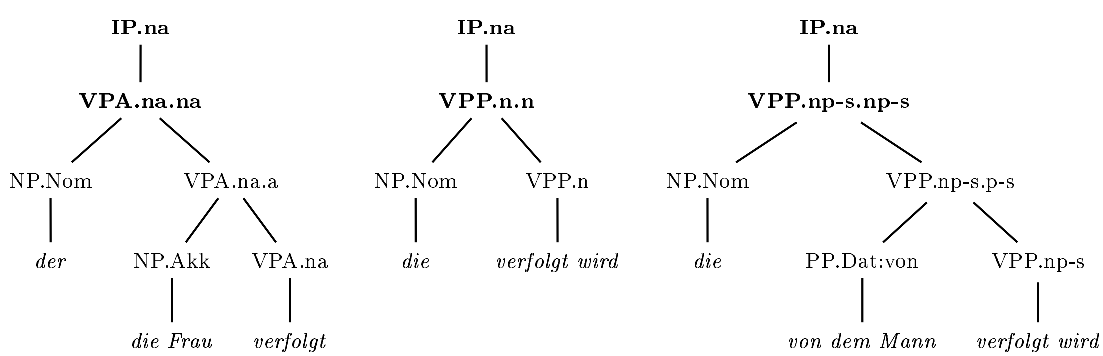


Figure 4: Generalising over the active-passive alternation of subcategorisation frames

Verb Phrase Types	Frame Types	Examples
VPA	n	*Sie schwimmt.*
	na	*Er sieht sie.*
	nd	*Er glaubt ihr.*
	np	*Die Entscheidung beruht auf einer guten Grundlage.*
	nad	*Sie verspricht ihm ein schönes Geschenk.*
	nap	*Sie hindert ihn am Stehlen.*
	ndp	*Er dankt ihr für ihr Verständnis.*
	ni	*Er versucht zu kommen.*
	di	*Ihm genügt, wenig zu verdienen.*
	nai	*Er hört sie ein Lied singen.*
	ndi	*Sie verspricht ihm zu kommen.*
	nr	*Sie fürchten sich.*
	nar	*Er erhofft sich Aufwind.*
	ndr	*Sie schließt sich der Kirche an.*
	npr	*Er hat sich als würdig erwiesen.*
	nir	*Sie stellt sich vor, eine Medaille zu gewinnen.*
VPP	n	*Er wird betrogen.*
	np-s	*Er wird von seiner Freundin betrogen.*
	d	*Ihm wird gehorcht.*
	dp-s	*Ihm wird von allen Leuten gehorcht.*
	p	*An die Vergangenheit wird appelliert.*
	pp-s	*Von den alten Leuten wird immer an die Vergangenheit appelliert.*
	nd	*Ihm wurde die Verantwortung übertragen.*
	ndp-s	*Ihm wurde die Verantwortung von seinem Chef übertragen.*
	np	*Sie wurde nach ihrem Großvater benannt.*
	npp-s	*Sie wurde von ihren Eltern nach ihrem Großvater benannt.*
	dp	*Ihr wird für die Komplimente gedankt.*
	dpp-s	*Ihr wird von ihren Kollegen für die Komplemente gedankt.*
	i	*Pünktlich zu gehen wurde versprochen.*
	ip-s	*Von den Schülern wurde versprochen, pünktlich zu gehen.*
	ni	*Er wurde verpflichtet, ihr zu helfen.*
	nip-s	*Er wurde von seiner Mutter verpflichtet, ihr zu helfen.*
	di	*Ihm wurde versprochen, früh ins Bett zu gehen.*
	dip-s	*Ihm wurde von seiner Freundin versprochen, früh ins Bett zu gehen.*
VPI	-	*zu schlafen*
	a	*ihn zu verteidigen*
	d	*ihr zu helfen*
	p	*an die Vergangenheit zu appellieren*
	r	*sich zu erinnern*
	ad	*seiner Mutter das Geschenk zu geben*
	ap	*ihren Freund am Gehen zu hindern*
	dp	*ihr für die Aufmerksamkeit zu danken*
	pr	*sich für ihn einzusetzen*
VPK	n	*Er bleibt Lehrer.*
	i	*Ihn zu verteidigen ist Dummheit.*

Table 2: Subcategorisation frame types

The basic verb frames without complements then collect the complements to the left. Consider for example the active transitive frame `VPA.na`:

```
VPA.na.na -> NP.Akk VPA.na.n'
VPA.na.na -> NP.Nom VPA.na.a'

VPA-RC.na.na -> RNP.Akk VPA.na.n'
VPA-RC.na.na -> RNP.Nom VPA.na.a'

VPA.na.n -> NP.Nom VPA.na'
VPA.na.a -> NP.Akk VPA.na'
```

In addition to NP-complements, prepositional phrases may be complements as well:

```
VPA.np.np -> PP.Dat:an VPA.np.n'
```

reflexive pronouns may be complements:

```
VPA.nr.nr -> PRF1.Akk VPA.nr.n'
```

and (saturated) infinitival clauses are possible extensions:

```
VPA.ni.ni -> VPI.max VPA.ni.n'
```

VPIs with maximal category label `VPI.max` generate saturated infinitival phrases, for example

```
VPI.max -> VPI.a.a'
VPI.max -> VPI.ad.ad'
```

Possible frames are listed in Table 2.

The leftmost NP- and PP-complements/-adjuncts for relative clauses compared to verb-final clauses are distinguished by naming them `RNP` and `RPP(ADJ)`, respectively, in order to identify their different syntactic characteristics.

On each level, adjuncts (prepositional phrases, adverbial chunks) are allowed, without changing the syntactic category, e.g.

```
VPA.na.n -> ADV1 VPA.na.n'
```

Verb Complexes

The active verb complexes are defined by a finite verb:

```
VPA -> VVFIN1'
```

or a past participle followed by a form of *sein* 'be' or *haben* 'have', e.g. *geschwommen ist* 'has swum', *gehört hatten* 'had heard':

```
VPA -> VVPP' VSFIN1
VPA -> VVPP' VHFIN1
```

or an infinitival main verb followed by an auxiliary form of *werden* 'become' or a modal, e.g. *gemalt haben wird* 'will have paint', *malen kann* 'can paint':

```
VPA -> VVINF1' VWFIN1
VPA -> VVINF1' VMFIN1
```

or an infinitival main verb with *zu* followed by a finite form of *haben*, e.g. *zu gehen hat* 'has to leave':

```
VPA -> VVIZU1' VHFIN1
```

or a finite form of *haben* followed by an infinitival main verb, e.g. *hat folgen können* 'could follow':

```
VPA -> VHFIN1 VVINF1'
```

The passive verb complexes are defined by a past participle form followed by a finite auxiliary or modal, e.g. *gemalt wird/worden ist* 'is/has been painted', *gemalt werden wird/kann* 'will/can be painted', *gemalt worden sein wird/kann* 'will/can have been painted':

```
VPP -> VVPP' VWFIN1
VPP -> VPPP' VSFIN1
VPP -> VPINF' VWFIN1
VPP -> VPINF' VMFIN1
VPP -> VPPP1' VWFIN1
VPP -> VPPP1' VMFIN
```

or a *zu*-infinitive followed by a finite form of *sein*, e.g. *zu antworten ist* 'to be answered':

```
VPP -> VVIZU1' VSFIN1
```

or a finite form of *haben* followed by an infinitival complex including a past participle, e.g. *hat gemalt werden können* 'could be painted':

```
VPP -> VHFIN1 VPINF1'
```

Infinitival verb complexes are bare infinitives or infinitives with *zu*:

```
VPI -> VVINF'
VPI -> VVIZU1'
```

Copula constructions consist of a predicative followed by a finite form of *sein, werden, bleiben* or the respective infinite form followed by a finite modal:

```
VPK -> PRED' VSFIN1
VPK -> PRED' VWFIN1
VPK -> PRED' VBFIN1
VPK -> PREDINF' VMFIN1
```

Predicatives can be nominative noun phrases (`NP.Nom`), prepositional phrases (`PP`), predicative adjectives (`ADJ1.Pred`), or adverbs (`ADV1`): *weil er Lehrer/in schlechtem Zustand/bekloppt/dort ist.*

4.3 Noun Chunks

On nominal categories, in addition to the four cases `Nom`, `Gen`, `Dat`, and `Akk`, case features with a disjunctive interpretation (such as `Dir` for `Nom` or `Akk`) are used. The grammar is written in such a way that non-disjunctive features are introduced high up in the tree. Figure 5 illustrates the use of disjunctive features in noun projections: the terminal `NN` contains the four-way ambiguous `Cas` case feature; the N-bar (`NN1`) and noun chunk `NC` projections disambiguate to two-way ambiguous case features `Dir` and `Obl`; the weak/strong (`Sw/St`) feature of `NN1` allows or prevents combination with a determiner, respectively; only at the noun phrase `NP` projection level, the case feature appears in disambiguated form.

The use of disjunctive case features results in some reduction in the size of the parse forest. Essentially the full range of agreement inside the noun phrase is enforced. Agreement between the subject `NP` and the tensed verb is not enforced by the grammar, in order to control the number of parameters and rules.

The noun chunk definition refers to Abney's chunk grammar organisation [Abney, 1996]: the noun chunk NC is a projection that excludes post-head complements and (adverbial) adjuncts introduced higher than pre-head modifiers and determiners, but includes participial pre-modifiers with their complements.

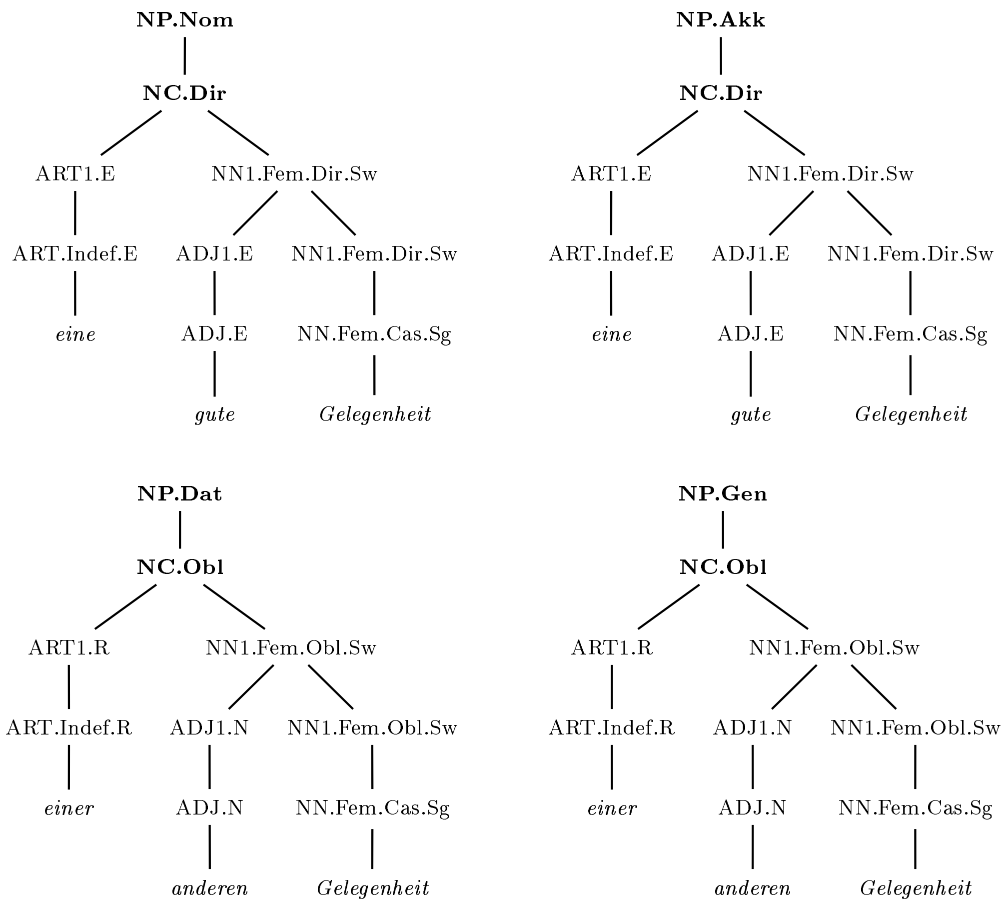


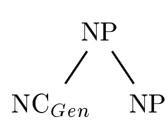
Figure 5: Noun projections

The noun phrases compose as follows: noun phrases generate noun chunks, possibly followed by a genitive modifier:

Examples: *die Mutter* / *die Mutter meines Freundes (in Italien)*

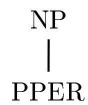
Prepositional, adverbial and genitive adjuncts are allowed on the phrase level:

Examples: *der Mann mit dem Hut* / *selbst meine Mutter*



Example: *des Schicksals schwieriger Weg (aus den Problemen)*

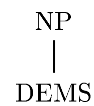
Personal pronouns, reflexive pronouns, possessive pronouns and demonstrative pronouns are directly generated by an NP, since no genitive modifiers are possible. The case value is inherited or inferred from the morphological ending of the pronoun.



Example: *ich, mir*

Example: *mich, sich*

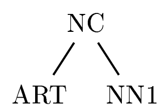
Example: *meiner, deins*



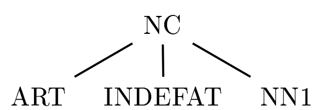
Example: *dieser, jenem*

Noun chunks project from the bar level, including determiners:

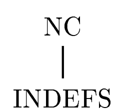
Example: *avancierten künstlerischen Manifestationen*



Example: *ein/solch/kein schlechter Verlierer*



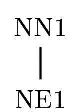
Example: *die paar Kröten*



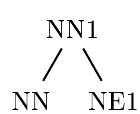
Example: *einige*

The bar level is generated from the terminal noun, a proper name, their combination, or a nominalised adjective:

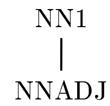
Example: *Orte*



Example: *Christoph Kolumbus*



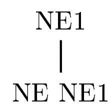
Example: *Eroberer Christoph Kolumbus*



Example: *Wichtigem*

Proper names consist of one or more names or an abbreviation of names:

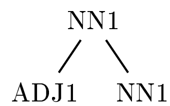
Example: *Jupp*



Example: *Christoph Kolumbus*

Example: *USSR*

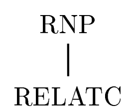
Adjectives are modifiers of the nominal bar level:



Example: *kleine Orte*

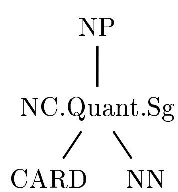
Noun phrases represented by relative pronouns are defined in a different way:

Example: *die*

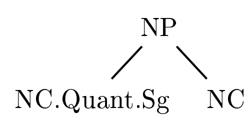


Example: *deren kräftig-warme Stimme*

Quantifying noun phrases are either composed as singular noun chunks –disregarding the plural-indicating cardinal–, possibly followed by the quantified element:

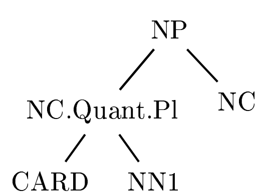


Example: *elf Glas*



Example: *elf Glas guten Saft*

or as plural noun chunks, necessarily followed by a noun chunk in the same case indicating the quantified element:



Example: *(mit) zwei Ladungen alten Tischen*

4.4 Prepositional Phrases

PPs are distinguished to indicate either a PP-argument or a PP-adjunct: arguments are identified by the label `PP`, adjuncts by `PPADJ`. For argument PPs, a restricted number of prepositions (*an, auf, gegen, in, über, vor, für, um, durch* for accusative PPs; *an, auf, in, über, aus, bei, mit, nach, von, vor, zu* for dative PPs) is accepted and added to the PP description, e.g. `PP.Akk:in`. The head of argument PPs is the preposition's adjacent node. Within adjunct PPs, only the case is encoded in the PP description (e.g. `PPADJ.Dat`), and the preposition represents the PP's head.

```
PP.Dat:mit -> APPR1.Dat:mit  NP.Dat'
PPADJ.Dat  -> APPR1.Dat:mit' NP.Dat
```

Prepositions (`APPR`) are marked by case (`Akk, Dat, Gen`) and –if they may occur in argument PPs, as in the above example– by the preposition itself.

The preposition selects a noun phrase, an adverbial chunk, a cardinal, or another prepositional phrase:

PP
APPR1 NP
Example: *an der Ecke*

PP
APPR1 ADV1
Example: *von hier*

PP
APPR1 CARD1
Example: *bis 1960*

PP
APPR1 PP

Example: *bis an das Ende*

Cardinals can represent prepositional phrases on their own:

PP.Dat
CARD

Example: *2003*

A prepositional phrase can combine with an adverbial chunk or be followed by a genitive noun phrase:

PP
ADV1 PP

Example: *schon auf der Hälfte*

PP
PP NP_{Gen}

Example: *auf dem Sofa der Kneipe*

A special case of prepositional phrases concerns those where the article and the preposition are represented within one morphological item. They combine with the nominal bar level instead of a noun phrase:

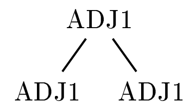
PPART
APPRART1 NN1

PPARTADJ
APPRART1 NN1

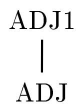
Examples: *im Haus, zur langen Nacht*

4.5 Adjectival Chunks

Adjective chunks consist of at least one, and possibly multiple adjectives. The morphological suffix is inherited.

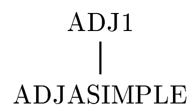


Example: *kleiner netter*



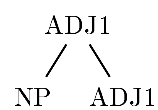
Examples: 1. *kleinem*, 2. *Schweizer* (without morphological feature)

Example: *viertes*

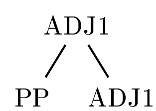


Example: *20.*

Bar level adjectives can subcategorise noun phrases or prepositional phrases:

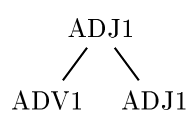


Example: *des Wartens müde*



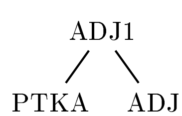
Example: *für den Einzelnen sehr günstiges*

They can be modified by adverbs:



Example: *relativ teure*

On the terminal level the particle *zu* can modify the adjective:

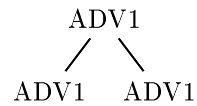


Example: *zu teure*

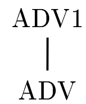
Predicative and adverbial adjuncts (`ADJ.Pred/Adv`) partly undergo the above definitions as well.

4.6 Adverbial Chunks

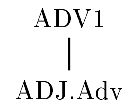
Adverb chunks also consist of one or more adverbs:



Example: *sehr langsam*



Example: *sehr*

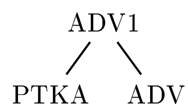


Example: *innen*

The adverb can be represented by the negation:

Example: *nicht*

The terminal category can be modified by the particle *zu*:



Example: *zu klein*

4.7 Determiner

Determiner `ART` are represented as definite `Def` or indefinite articles `Indef`, possessive pronouns `POSAT`, demonstrative pronouns `DEMAT`, indefinite pronouns `INDEFAT` or a combination. The are marked for their difference in definiteness and/or by a morphological suffix.

Examples:

```
ART1.Def.M   -> ART.Def.M'
ART1.Indef.E -> ART.Indef.E'
ART1.Indef.M -> POSAT.M'
ART1.Def.N   -> DEMAT.N'
ART1.Def.E   -> INDEFAT.E'
ART1.Indef.E -> INDEFAT.E'
ART1.Indef.E -> INDEFAT.Z' ART1.Indef.E
```

4.8 Coordination

Coordination rules are those which blow up the number of grammar rules enormously. Therefore, only few types of coordination were realised:

- On the clause level, all IP-types are combinable.
- Non-maximally saturated active and passive verb phrases can be coordinated in case only one (the same) argument is missing, e.g.

  `VPA-RC.na.na -> RNP.Akk VPA.nad.nd KON1 VPA.na.n'`

  Verbal complexes can be coordinated presupposing they are in the same mode. Identical auxiliaries and modals are combinable as well.
- Noun phrases with identical case can be coordinated.
- Prepositional phrases with identical case and prepositional head can be coordinated.
- Cardinals, adverbial and adjectival chunks, subordinating conjunctions and truncated structures for nouns and adjectives on the bar level can be coordinated.

4.9 Untagged Words

On the terminal level, nouns which are not recognised by our morphological analyser are assigned auxiliary tags by the parser (cf. Section 5.1.4) which are themselves dominated by all respective terminal noun/proper name terminal categories:

```
NN.Masc.Cas.Sg -> UNTAGGED-NN'
NN.Masc.Cas.Pl -> UNTAGGED-NN'
NN.Fem.Cas.Sg  -> UNTAGGED-NN'
NN.Fem.Cas.Pl  -> UNTAGGED-NN'
NN.Neut.Cas.Sg -> UNTAGGED-NN'
NN.Neut.Cas.Pl -> UNTAGGED-NN'
NE.NoGend.Cas.Sg -> UNTAGGED-NE'
NE.NoGend.Cas.Pl -> UNTAGGED-NE'
```

5 Statistical Grammar Training

The context-free grammar rules were assigned random frequencies, on which basis a probabilistic context-free grammar could be created. The probabilistic grammar was then trained with the head-lexicalised probabilistic context-free parser `LoPar` [Schmid, 2000]. The parser is an implementation of the Left-Corner algorithm for parsing and of the Inside-Outside algorithm [Lari and Young, 1990] –an instance of the Expectation-Maximisation (EM) algorithm [Baum, 1972]– for parameter estimation.

The resulting grammar model is a trained, head-lexicalised probabilistic version of the original context-free grammar [Carroll, 1995, Carroll and Rooth, 1998], including lexicalised model parameters: it contains lexicalised rules, i.e. grammar rules referring to a specific lexical head, and lexical choice parameters, a measure of lexical coherence between lexical heads. Concerning verbs, for example, the lexical rule parameters serve as basis for probability distributions over subcategorisation frames, and the lexical choice parameters supply us with nominal heads of subcategorised noun phrases, as basis for selectional constraints.

For a detailed description of (head-lexicalised) probabilistic context-free grammars and the parser's design and implementation see [Schmid, 2000].

What is the linguistically optimal strategy for training a head-lexicalised probabilistic context-free grammar, i.e. estimating the model parameters in the optimal way? The EM-algorithm guarantees improving an underlying model towards a (local) maximum of the likelihood of the training corpus, but is that adequate for improving the linguistic representation within the probabilistic model? Various training strategies have been developed in the past years, with preliminary results referred to by [Beil et al., 1999].
Elaborating the optimal training strategy results from the interaction between the linguistic and mathematical motivation and properties of the probability model:

- Mathematical motivation: perplexity of the model

  The *perplexity* $Perp_M(C)$ of a corpus C wrt. a language model M is a measure of fit for the model. The perplexity is defined as

  $$Perp_M(C) = e^{\frac{-log P_M(C)}{N}}$$

  where $P_M(C)$ is the *likelihood* of corpus C according to model M, and N is the size of the corpus. Intuitively, the perplexity measures the uncertainty about the next word in a corpus. For example, if the perplexity is 23, then the uncertainty is as high as it is when we have to choose from 23 alternatives of equal probability.

  The perplexity on the training and test data should decrease during training. At one point the perplexity on the test data will increase again which is referred to as *over-training*. The optimal point of time to stop the training is at the minimum of perplexity, before the increase.

- Linguistic motivation: representation of linguistic features

  The linguistic parameters can be controlled by investigating rule and lexical choice parameters, e.g. what is the probability distribution over subcategorisation frames concerning the verb *achten* (ambiguous between 'to respect' and 'to pay attention'), and does it correlate to existing lexical information?

  In addition, the model were inspected by controlling the parsing performance on specified grammatical structures, i.e. noun chunks and verb phrases have been assigned labels which form the basis for evaluating parses.

Section 5.1 describes the setup of the training environment. Section 5.2 refers to the up to the present optimal training strategy. In Section 5.3 the resulting model is evaluated; Section 5.4 describes the linguistic performance in more detail, i.e. strength and weaknesses of the model are investigated.

5.1 Training Environment

Minimum demands for `LoPar` include a grammar, a file referring to possible start symbols, a lexicon, and open class definitions for capitalised and non-capitalised unknown words. In addition, pooling classes can be defined.

The obligatory file names need identical prefixes `<file>`, followed by the respective suffixes `.gram`, `.start`, `.lex`, and `.OC/.oc`.

5.1.1 Grammar

The grammar rules contain a frequency and the context-free grammar rule itself, for example:

```
5674.59 VPA.na.na NP.Nom VPA.na.a'
```

with the syntactic category in the first column as parent category of the rule. The head needs to be marked by `'`.

5.1.2 Start Symbols

This file contains the probabilities of the start symbols in the grammar, for example

```
printf 'S\t10\n' > lopar.start
```

if only one start symbol `S` appears in the grammar.

5.1.3 Lexicon

The lexicon contains one lexical entry per line, starting with a word-form followed by a tab and a sequence of pos/frequency pairs, for example

```
Versuchen   NN.Masc 1   VVFIN 1
```

For parsing with stems the stems have to be inserted into the lexicon as well, for example:

```
Versuchen   NN.Masc 1 Versuch   VVFIN 1 versuchen
```

5.1.4 Open Classes

The files contain the open classes within the grammar which by default are assigned to unknown words. `LoPar` distinguishes capitalised and non-capitalised (unknown) words and therefore demands two separate files, `<file>.OC` and `<file>.oc`. The format in both files is one class and frequency per line, for example

```
VVFIN 1
```

You can also create an empty file:

```
touch <file>.oc
```

I used the following open classes:

Capitalised Words (cf. the grammar rules in Section 4.9)

```
UNTAGGED-NE    50
UNTAGGED-NN    10
```

Non-Capitalised Words

```
ADJ.E          10
ADJ.M          10
ADJ.N          10
ADJ.R          10
ADJ.S          10
ADJ.Pred       10
VVPP            1
VVFIN           1
VVINF           1
VVIZU           1
```

5.1.5 Parameter Pooling

For parameter pooling of the lexical choice parameters in lexicalisation and lexicalised training, the relevant pooling files for pooling parent categories and/or pooling child categories needed to be defined. Each line in the files corresponds to one pooling class, containing the class name followed by its members, for example

`VPA-n VPA-RC.n.n VPA.n VPA.n.n`

I defined parent pooling classes for all VPs, NPs and nominal terminals, and child pooling classes for adjectives and nominal terminals.

5.2 Training Strategy

For training the model parameters we used 90% of the corpora, i.e. 90% of the verb-final and 90% of the relative clauses, a total of 1.4 million clauses. Every 10th sentence was cut out of the corpora to generate a test corpus. The training was performed in the following steps:

1. Initialisation:

   The grammar was initialised by identical frequencies for all context-free grammar rules.

   Comparative initialisations with random frequencies had no effect on the model development.

2. Unlexicalised training:

   The training corpus was parsed once with `LoPar`, re-estimating the frequencies twice.

   The optimal training strategy proceeds with few parameter re-estimations. Without re-estimations or with a large number of re-estimations the model was effected to its disadvantage.

   With less unlexicalised training more changes during lexicalised training take place later on.

3. Lexicalisation:

   The unlexicalised model was turned into a lexicalised model by

   - setting the probabilities of the lexicalised rule probabilities to the values of the respective unlexicalised probabilities
   - initialising the lexical choice and lexicalised start probabilities uniformly.

4. Lexicalised training:

   Three training iterations were performed on the training corpus, re-estimating the frequencies after each iteration.

   Comparative numbers of iterations (up to 40 iterations) showed that more iterations of lexicalised training did not have further effect on the model.

To achieve a reduction of parameters and improve the lexical choice model, we utilised parameter pooling as described in Section 5.1.5: all active, passive and non-finite verb frames were pooled according to shared arguments, disregarding the saturation state of the frames, in order to generalise over their arguments without taking into account their positional facilities. In addition, each of the categories describing noun phrases, noun chunks, the noun bar level and proper names was pooled disregarding the features for gender, case and number, thus allowing to generalise over open class categories like adjectives which combine with nouns disregarding the features.

5.3 Probability Model Evaluation

As mentioned above, main background for the development of the training strategy were the perplexity of the model as the measure of mathematical evaluation on the one hand, and the parsing accuracy of grammatical structures as the measure of linguistic evaluation on the other hand. Figure 6 displays the development of the perplexity on the training data, Figure 7 the development of the perplexity on the test data, both referring to the experiment described in Section 5.2, illustrating lexicalised training up to its fifth iteration.

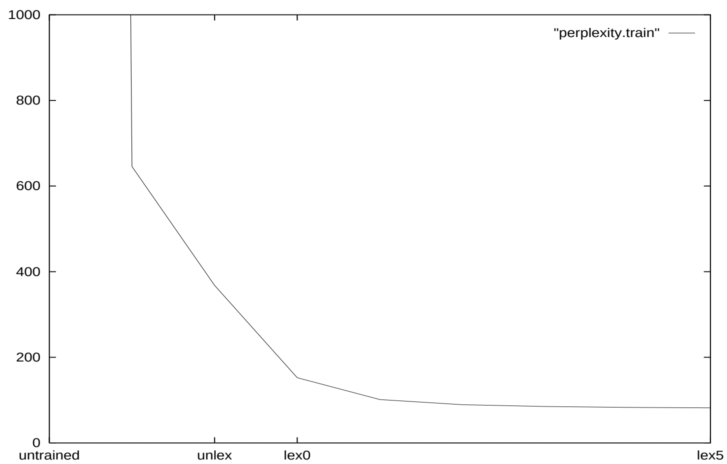


Figure 6: Perplexity on training data

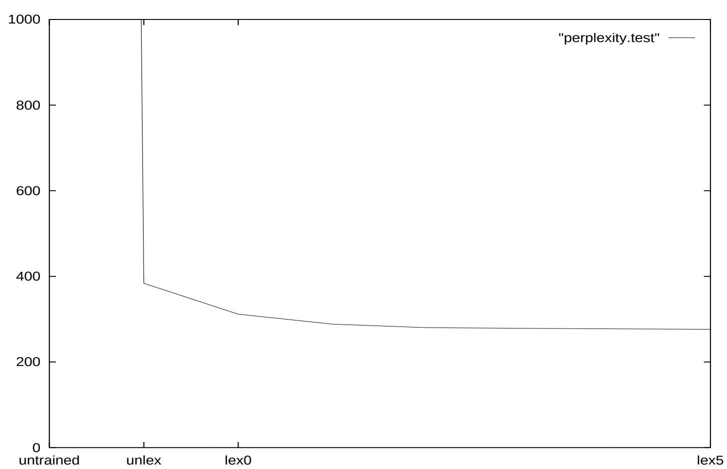


Figure 7: Perplexity on test data

As the figures show, both the perplexity on the training data and the perplexity on the test data monotonously decrease during training, which means that according to perplexity the model improves steadily and has not reached the status of over-training yet.

The linguistic parameters of the models were evaluated concerning the identification of noun chunks and subcategorisation frames. We randomly extracted 200 relative clauses and 200 verb-final clauses from the test data and hand-annotated them with the relevant syntactic categories, the relative clauses with noun chunk labels, and all clauses with frame labels. In addition, we extracted 100 randomly chosen relative clauses for each of the six verbs *beteiligen* 'participate', *erhalten* 'receive', *folgen* 'follow', *verbieten* 'forbid', *versprechen* 'promise', *versuchen* 'try', and hand-annotated them with their subcategorisation frames. Probability models were evaluated by making the models determine the Viterbi parses (i.e. the most probable parses) of the test data, extracting the categories of interest (i.e. noun chunks and subcategorisation frame types) and comparing them with the annotated data. The noun chunks were evaluated according to

- the range of the noun chunks, i.e. did the model find a chunk at all?
- the range and the identifier of the noun chunks, i.e. did the model find a noun chunk and identify the correct syntactic category and case?

and the subcategorisation frames were evaluated according to the frame label, i.e. did the model determine the correct subcategorisation frame for a clause? Precision was measured in the following way:[2]

$$precision = \frac{tp}{tp + fp}$$

Figures 8 and 9 present the strongly different development of noun chunk and subcategorisation frame representations within the models, ranging from the untrained model until the fifth iteration of lexicalised training. Noun chunks were modelled sufficiently by an unlexicalised trained grammar, lexicalisation made the modelling worse. Verb phrases in general needed a combination of unlexicalised and lexicalised training, but the representation strongly depended on the specific item. Unlexicalised training advanced frequent phenomena (compare, for example, the representation of the transitive frame with direct object for *erfahren* and with indirect object for *folgen*), lexicalisation and lexicalised training improved the lexicalised properties of the verbs, as expected.

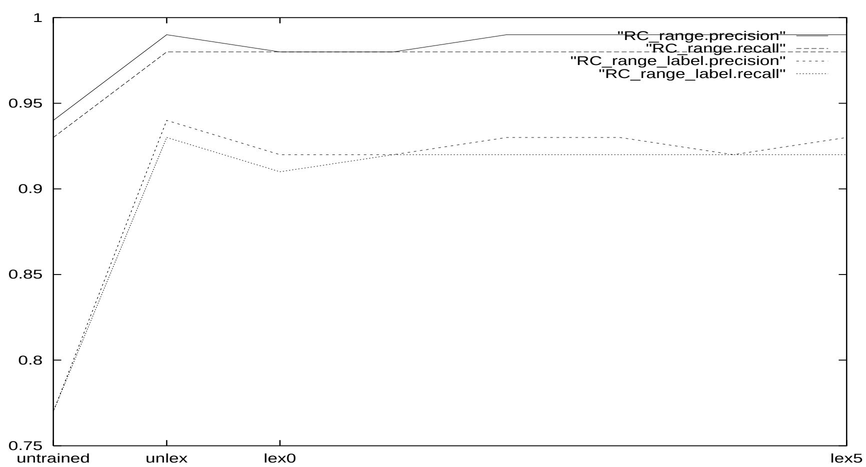


Figure 8: Development of precision and recall values on noun chunk range and label

It is obvious that perplexity can hardly measure the linguistic performance of the training strategy and resulting models; the perplexity (on training as well as on test data) is a monotonously decreasing curve, but as explained above the linguistic model performance develops differently according to different phenomena. So perplexity can only serve as rough indicator whether the model reaches towards an optimum, but linguistic evaluation determines the optimum.

2 - tp: identified chunk/label is correct
- fp: identified chunk/label is not correct

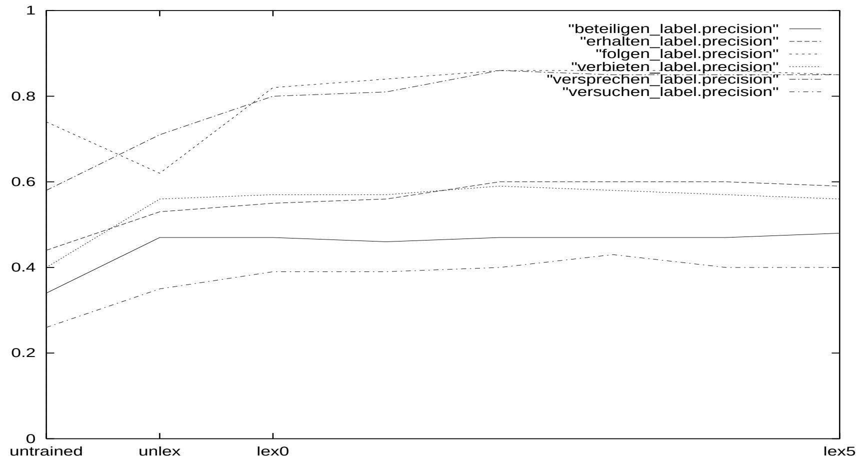


Figure 9: Development of precision values on subcategorisation frames for specific verbs

The precision values of the "best" model according to the training strategy in Section 5.2 were as in Table 3.

Noun Chunks		Subcategorisation Frames							
range	range+label	rc	vfinal	*beteiligen*	*erhalten*	*folgen*	*verbieten*	*versprechen*	*versuchen*
98%	92%	63%	73%	48%	61%	88%	59%	80%	49%

Table 3: Precision values on noun chunks and subcategorisation frames

For comparison reasons, we evaluated the subcategorisation frames of 200 relative clauses extracted from the training data. Interestingly, there were no striking differences concerning the precision values.

Without utilising the pooling option the precision values for low-frequent phenomena such as non-finite frame recognition was worse, e.g. the precision for the verb *versuchen* was 9% less than with pooling.

5.4 Investigating the Linguistic Performance of the Model

Which linguistic aspects could be learned by the probability model, i.e. what is the strength and what are the weaknesses of the model? Noun chunks, subcategorisation frames and prepositional frames have been investigated.

Concerning the noun chunks, a remarkable number was identified correctly, concerning their structure (i.e. what is a noun chunk) as well as their category (i.e. which case is assigned to the noun chunk). Before training, a large number of noun chunks was assigned wrong case, but after training the mistakes were mostly corrected except for few noun chunks being assigned the accusative case instead of nominative or dative.

For subcategorisation frames, the distribution and confusion of the multiple frames is manifold. Some interesting feature developments are cited below.

- Highly common subcategorisation types such as the transitive frame are learned in unlexicalised training and then slightly unlearned in lexicalised training. Less common subcategorisation types such as the demand for an indirect object are unlearned in unlexicalised training, but improved during lexicalised training.

- It is difficult and was not effectively learned to distinguish between prepositional phrases as verbal complements and adjuncts.

- The active present perfect verb complexes and passive of condition were confused, because both are composed by a past participle and a form of *be*, e.g. *geschwommen ist* 'has swum' vs. *gebunden ist* 'is bound'.

- Copula constructions and passive of condition were confused, again because both may be composed by a past participle and a form of *to be*, e.g. *verboten ist* 'is forbidden' vs. *erfahren ist* 'is experienced'.

- Noun chunks belonging to a subcategorised non-finite clause were partly parsed as arguments of the main verb. For example, *der ihn zu überreden versucht* 'who him_{acc} tried to persuade' was parsed as demanding an accusative plus a non-finite clause instead of recognising that the accusative object is subcategorised by the embedded infinitival verb.

- Reflexive pronouns appeared in the subcategorisation frame as either reflexive pronoun itself or as accusative or dative noun chunk. The correct or wrong choice of frame type containing the reflexive pronoun was consequently right or wrong for different verbs. For example, the verb *sich befinden* 'to be situated' was generally parsed as a transitive, not as inherent reflexive verb.

This feature confusion reflects the background for the identification of the frame types concerning the specifically chosen verbs:

- The verb *beteiligen* was mostly parsed as transitive verb. Two sources of mistakes were combined here: (i) the verb was assigned a transitive instead of inherent reflexive frame, and (ii) the obligatory prepositional phrase was consequently parsed as adjunct instead of argument. All feature tendencies were already determined by unlexicalised training and not corrected in lexicalised training.

- The transitive frame of *erhalten* was recognised well, not many mistakes were made except for the PP-assignment.

- As consequence of unlexicalised training, the verb *folgen* was partly parsed as transitive, but lexicalised training corrected that tendency.

- The main problem for the verb *verbieten* was being assigned a copula-construction instead of a passive of condition.

- For the verb *versprechen* the main mistake was using the dominance of the bitransitive frame also for parsing the transitive reflexive verb *sich versprechen*.

- The main mistake for *versuchen* was parsing a direct object instead of recognising the object's correlation with the embedded infinitival verb.

We conclude the linguistic feature description by presenting probability distributions of selected verbs over subcategorisation frames in Table 4[3], as extracted by questioning tools on the model parameters.

[3]Examples are only given in case the frame usage is possible. Otherwise an explanation for a wrong frame indication is given.

Verb	Prob.	Frame	Example
funktionieren	79%	IP.n	*weil die Maschine funktioniert*
	29%	IP.np	[PP cannot be argument]
erfahren	50%	IP.na	*weil er die Neuigkeit erfahren hat*
	25%	IP.np	*weil er von den Änderungen erfahren will*
	11%	IP.n	[intransitive use not possible]
	10%	IP.nap	[PP cannot be argument]
folgen	67%	IP.nd	*weil er ihr folgen wollte*
	13%	IP.n	*weil wichtige Entscheidungen folgen werden*
erlauben	42%	IP.na	*weil meine Eltern vieles erlaubt haben*
	29%	IP.nad	*weil sie mir vieles erlaubt haben*
achten	45%	IP.np	*weil das Kind auf die Ampel achten sollte*
	31%	IP.na	*daß wir die Bemühungen achten*
	19%	IP.n	[intransitive use not possible]
basieren	89%	IP.np	*daß die Ausnahme auf der Regel basiert*
beginnen	48%	IP.np	*daß wir mit der Schule beginnen möchten*
	24%	IP.n	*daß die Vorlesung beginnt*
	11%	IP.na	*weil wir das Frühstück bereits begonnen haben*
scheinen	32%	IP.ni	*weil die Regelung zu funktionieren scheint*
	25%	IP.n	*weil die Sonne heute scheint*
	16%	IP.nai	[accusative should be parsed as direct object of embedded infinitival verb]
erweisen	61%	IP.nr	[PP as argument needed]
	17%	IP.npr	*weil sie sich als eine gute Fee erwiesen hat*
	11%	IP.nad	*weil er ihr die Ehre erweist*
enden	66%	IP.np	*weil die Stunde mit dem Glockenschlag enden wird*
	29%	IP.n	*weil die schönsten Zeiten enden werden*
beteiligen	48%	IP.npr	*weil wir uns an dem Kauf beteiligen wollen*
	22%	IP.np	[confusion copula construction and passive of condition]
	15%	IP.nr	[PP as argument needed]

Table 4: Probability distribution over subcategorisation frames

6 Linguistic Exploitation of the Statistical Grammar Model

Having trained the statistical grammar models, there exists valuable lexical information. But how to detect it? What are the possibilities to determine relevant lexical information and apply it to interesting tasks? The following sections refer to the potential of the grammar models, with Section 6.1 presenting a collection of lexicalised probabilities for verbs; Section 6.2 applies Viterbi parsing on basis of the lexical probabilities to an example sentence, followed by Section 6.3 extracting an empirical database of subcategorisation frames from Viterbi parses.

The information can be used straightly as lexical description, or as input for lexicon tools, such as semantic clustering techniques [Rooth et al., 1999, Schulte im Walde, 2000], or as basis for a variety of applications, e.g. parser improvement [Riezler et al., 2000], machine translation [Prescher et al., 2000], or chunking [Schmid and Schulte im Walde, 2000].

6.1 Lexicalised Probabilities

The model parameters can be queried by tools. First, we queried for the subcategorisation frames of specific verbs. This kind of parameter belongs to the lexicalised rules; it specifies the probability of the sentence generating the category `IP.<Frame>`, depending on a verb. Following you find the relevant probabilities of the IPs, for display reasons with a cut-off probability of 10%:

```
Verb: glauben
---------------------------------
prob         IP.<frame>
---------------------------------
0.45115      IP.n
0.14787      IP.na
0.13740      IP.np

Verb: folgen
---------------------------------
prob         IP.<frame>
---------------------------------
0.70054      IP.nd
0.13717      IP.n

Verb: achten
---------------------------------
prob         IP.<frame>
---------------------------------
0.45376      IP.np
0.30238      IP.na
0.18469      IP.n
```

```
Verb: geben
---------------------------------
prob         IP.<frame>
---------------------------------
0.51598      IP.na
0.22681      IP.nap
0.15378      IP.nad

Verb: enden
---------------------------------
prob         IP.<frame>
---------------------------------
0.66980      IP.np
0.28282      IP.n

Verb: beteiligen
---------------------------------
prob         IP.<frame>
---------------------------------
0.52067      IP.npr
0.18734      IP.np
0.14666      IP.nr
```

Secondly, we queried for the probabilities of subcategorised prepositional phrases in verb phrases (containing a prepositional phrase as one argument). The probabilities also represent a kind of lexicalised rule parameters: the probability of a certain PP, e.g. a PP with dative case and headed by the preposition *mit*, representing the subcategorised PP in the subcategorisation frame, e.g. the frame `np`.

```
Verb: sprechen   VP: VPA.np
---------------------------------------
prob         rule
---------------------------------------
0.18752      PP.Dat:von
0.13271      PP.Akk:für
0.13136      PP.Dat:mit

Verb: enden   VP: VPA.np
---------------------------------------
prob         rule
---------------------------------------
0.25152      PP.Dat:mit
0.22102      PP.Dat:in
0.20671      PP.Dat:an

Verb: eignen   VP: VPA.npr
---------------------------------------
prob        rule
---------------------------------------
0.39232     PP.Akk:für
0.15285     PP.Dat:zu
```

In the final example, I filtered frequency distributions over nominal heads in subcategorised noun phrases. This kind of parameter belongs to the lexical choice parameters; it specifies the probability of a certain lemma, e.g. the noun *Kind* 'child', as head of a subcategorised noun phrase, e.g. an NP with accusative case.

```
Verb: drohen   VP: VPA.nd -- NP.Nom
---------------------------------------
freq      word
---------------------------------------
18.9      Gefahr
17.0      Abschiebung
17.0      Verfolgung
13.8      Todesstrafe
 7.9      Tod
 5.0      Arbeitslosigkeit
 5.0      Ausweisung
 5.0      Entlassung
 5.0      Kündigung

Verb: erziehen   VP: VPA.na -- NP.Akk
---------------------------------------
freq      word
---------------------------------------
16.0      Kind
 2.0      Junge
 2.0      Sohn
 2.0      Tochter

Verb: entstammen   VP: VPA.nd -- NP.Dat
---------------------------------------
freq      word
---------------------------------------
3.0       Familie
3.0       Jahrhundert
3.0       Welt
2.0       Disziplin
2.0       Drogenhandel
2.0       Elternhaus
2.0       Zeit
```

6.2 Viterbi Parses

With `LoPar`, it is possible to parse a corpus unambiguously by selecting the respective analysis with the highest probability (called *Viterbi parse*). Viterbi parses are printed in a list notation; graphical tools allow the parse tree representation. For example, the Viterbi parse of the relative clause *die vielen Menschen das Leben retten könnte* 'which could save many people's lives' is represented by the parse tree in Figure 10. The parser correctly chose the ditransitive subcategorisation frame `nad` for the verb *retten* 'save', and provided the relevant NPs with the correct case, *die* as a nominative relative pronoun, *vielen Menschen* as an NP with dative case, and *das Leben* as an NP with accusative case. Viterbi parsing is used to build large parsed corpora (called *treebanks*), or as an intermediate step in larger NLP systems for e.g. machine translation, text mining, information retrieval, question answering, query analysis.

6.3 Empirical Subcategorisation Frame Database

Section 6.2 introduced Viterbi parses as a method for determining the most probable parse of a sentence. I collected the parses to build an empirical database, as input to complex NLP systems. The database has actually been used for semantic clustering (cf. [Rooth et al., 1999, Schulte im Walde, 2000]) and experiments on verb biases concerning lexical syntactic preferences [Lapata et al., 2001].

The following lines represent some example subcategorisation frame tokens for German. The examples start with a verb-final clause, followed by all arguments and the verb frame.

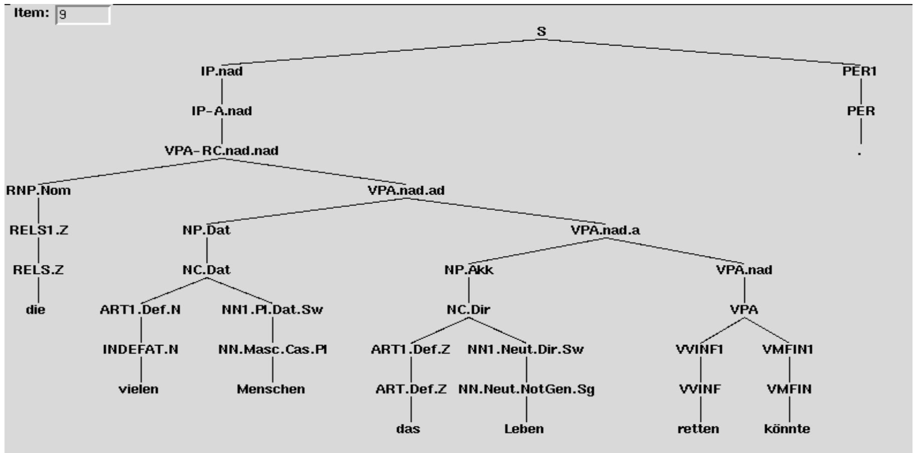


Figure 10: Viterbi parse

```
S       dass in diesem Jahr der grosse Coup gelingen würde.
NP.Nom  Coup
IP.n    gelingen

S       weil die Stadtväter Schmiergelder für die Einrichtung
        eines modernen Müllplatzes einsteckten.
NP.Nom  Stadtväter
NP.Akk  Schmiergelder
IP.na   einsteckten

S       dass diese Kunst unverfälschten menschlichen Bedürfnissen entspricht.
NP.Nom  Kunst
NP.Dat  Bedürfnissen
IP.nd   entspricht
```

References

[Abney, 1996] Abney, S. (1996). Chunk Stylebook. Technical report, Seminar für Sprachwissenschaft, Universität Tübingen.

[Baum, 1972] Baum, L. E. (1972). An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. *Inequalities*, III:1–8.

[Beil et al., 1999] Beil, F., Carroll, G., Prescher, D., Riezler, S., and Rooth, M. (1999). Inside-Outside Estimation of a Lexicalized PCFG for German. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, College Park, MD.

[Carroll, 1995] Carroll, G. (1995). *Learning Probabilistic Grammars for Language Modeling*. PhD thesis, Department of Computer Science, Brown University.

[Carroll, 1997] Carroll, G. (1997). *Manual Pages for `charge`, `hyparCharge`*. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

[Carroll and Rooth, 1998] Carroll, G. and Rooth, M. (1998). Valence Induction with a Head-Lexicalized PCFG. In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing*, Granada, Spain.

[Christ, 1994] Christ, O. (1994). *The IMS Corpus Workbench*. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

[Christ et al., 1999] Christ, O., Schulze, B. M., Hofmann, A., and König, E. (1999). *The IMS Corpus Workbench: Corpus Query Processor*. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

[Lapata et al., 2001] Lapata, M., Keller, F., and Schulte im Walde, S. (2001). Verb Frame Frequency as a Predictor of Verb Bias. *Journal of Psycholinguistic Research*.

[Lari and Young, 1990] Lari, K. and Young, S. J. (1990). The Estimation of Stochastic Context-Free Grammars using the Inside-Outside Algorithm. *Computer Speech and Language*, 4:35–56.

[Prescher et al., 2000] Prescher, D., Riezler, S., and Rooth, M. (2000). Using a Probabilistic Class-Based Lexicon for Lexical Ambiguity Resolution. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken.

[Riezler et al., 2000] Riezler, S., Prescher, D., Kuhn, J., and Johnson, M. (2000). Lexicalized Stochastic Modeling of Constraint-Based Grammars using Log-Linear Measures and EM Training. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00)*, Hong Kong.

[Rooth et al., 1999] Rooth, M., Riezler, S., Prescher, D., Carroll, G., and Beil, F. (1999). Inducing a Semantically Annotated Lexicon via EM-Based Clustering. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, Maryland.

[Schiller and Stöckert, 1995] Schiller, A. and Stöckert, C. (1995). *DMOR*. Insitut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

[Schmid, 1999] Schmid, H. (1999). *YAP: Parsing and Disambiguation with Feature-Based Grammars*. PhD thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

[Schmid, 2000] Schmid, H. (2000). Lopar: Design and Implementation. Arbeitspapiere des Sonderforschungsbereichs 340 *Linguistic Theory and the Foundations of Computational Linguistics* 149, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

[Schmid and Schulte im Walde, 2000] Schmid, H. and Schulte im Walde, S. (2000). Robust German Noun Chunking with a Probabilistic Context-Free Grammar. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-00)*, pages 726–732, Saarbrücken, Germany.

[Schulte im Walde, 2000] Schulte im Walde, S. (2000). Clustering Verbs Semantically According to their Alternation Behaviour. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-00)*, pages 747–753, Saarbrücken, Germany.

[Schulze, 1996] Schulze, B. M. (1996). *GermLem – ein Lemmatisierer für deutsche Textcorpora*. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.