Chapter 3

Statistical Grammar Model

This chapter describes the implementation, training and lexical exploitation of a German statistical grammar model. The model provides empirical lexical information, specialising on but not restricted to the subcategorisation behaviour of verbs. It serves as source for the German verb description at the syntax-semantic interface, which is used within the clustering experiments.

Before going into the details of the grammar description I introduce the definition of subcategorisation as used in the German grammar. The subcategorisation of the verbs distinguishes between obligatory and facultative verb complements.[1] The subcategorisation is defined by the arguments of a verbs, i.e. only obligatory complements are considered. A problem arises, because both in theory and in practice there is no clear-cut distinction between arguments and adjuncts. (a) Several theoretical tests have been proposed to distinguish arguments and adjuncts on either a syntactic or semantic basis, cf. Schütze (1995, pages 98–123) for an overview of such tests for English. But different tests have different results with respect to a dividing line between arguments and adjuncts, so the tests can merely be regarded as heuristics. I decided to base my judgement regarding the argument-adjunct distinction on the optionality of a complement: If a complement is optional in a proposition it is regarded as adjunct, and if a complement is not optional it is regarded as argument. I am aware that this distinction is subjective, but it is sufficient for my needs. (b) In practice, a statistical grammar would never learn the distinction between arguments and adjuncts in a perfect way, even if there were theoretically exact definitions. In this sense, the subcategorisation definition of the verbs in the German grammar is an approximation to the distinction between obligatory and facultative complements.

The chapter introduces the theoretical background of lexicalised probabilistic context-free grammars (Section 3.1) describes the German grammar development and implementation (Section 3.2), and the grammar training (Section 3.3). The empirical lexical information in the resulting statistical grammar model is illustrated (Section 3.4), and the core part of the verb information, the subcategorisation frames, are evaluated against manual dictionary definitions (Section 3.5).

[1] I use the term *complement* to subsume both arguments and adjuncts, and I refer to *arguments* as obligatory complements and *adjuncts* as facultative complements.

3.1 Context-Free Grammars and their Statistical Extensions

> *At one level of description, a natural language is a set of strings – finite sequences of words, morphemes, phonemes, or whatever.*
>
> Partee, ter Meulen, and Wall (1993, page 431)

Regarding natural language as a set of strings, a large part of language structures can be modelled using context-free descriptions. For that reason, context-free grammars have become a significant means in the analysis of natural language phenomena. But context-free grammars fail in providing structural and lexical preferences in natural language; therefore, a probabilistic environment and a lexicalisation of the grammar framework are desirable extensions of the basic grammar type.

This section describes the theoretical background of the statistical grammar model: Section 3.1.1 introduces context-free grammars, Section 3.1.2 introduces probabilistic context-free grammars, and Section 3.1.3 introduces an instantiation of lexicalised probabilistic context-free grammars. Readers familiar with the grammar formalisms might want to skip the respective parts of this section.

3.1.1 Context-Free Grammars

Context-free grammars can model the most natural language structure. Compared to linear language models –such as n-grams– they are able to describe recursive structures (such as complex nominal phrases).

Definition 3.1 *A context-free grammar CFG is a quadruple $\langle N, T, R, S \rangle$ with*

N	*finite set of non-terminal symbols*
T	*finite set of terminal symbols,* $T \cap N = \emptyset$
R	*finite set of rules* $C \rightarrow \gamma$, $C \in N$ *and* $\gamma \in (N \cup T)^*$
S	*distinguished start symbol,* $S \in N$

As an example, consider the context-free grammar in Table 3.1. The grammar unambiguously analyses the sentences *John loves Mary* and *John loves ice-cream* as represented in Figure 3.1. If there were ambiguities in the sentence, the grammar would assign multiple analyses, without defining preferences for the ambiguous readings.

N	S, NP, PN, CN, VP, V
T	*John*, *Mary*, *ice-cream*, *loves*
R	S $\rightarrow$ NP VP,
	NP $\rightarrow$ PN,
	NP $\rightarrow$ CN,
	VP $\rightarrow$ V NP,
	PN $\rightarrow$ *John*,
	PN $\rightarrow$ *Mary*,
	CN $\rightarrow$ *ice-cream*,
	V $\rightarrow$ *loves*
S	S

Table 3.1: Example CFG

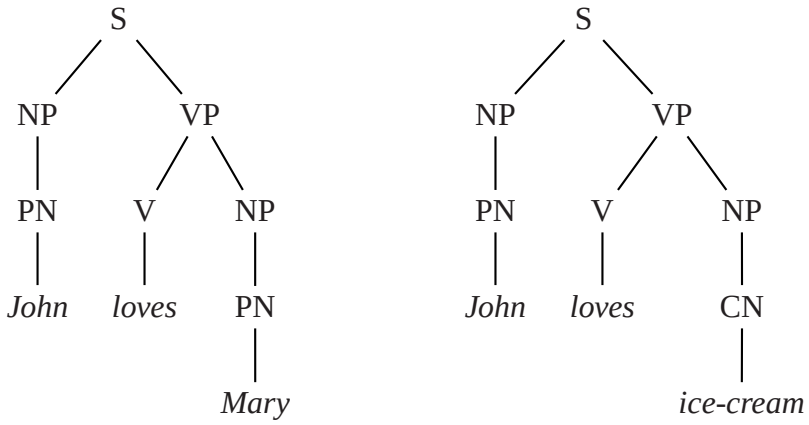


Figure 3.1: Syntactic analyses for *John loves Mary* and *John loves ice-cream*

The example is meant to give an intuition about the linguistic idea of context-free grammars. For details about the theory of context-free grammars and their formal relationship to syntactic trees, the reader is referred to Hopcroft and Ullman (1979, chapter 4) and Partee *et al.* (1993, chapter 16).

To summarise, context-free grammars can model the a large part of natural language structure. But they cannot express preferences or degrees of acceptability and therefore cannot resolve ambiguities.

3.1.2 Probabilistic Context-Free Grammars

Probabilistic context-free grammars (PCFGs) are an extension of context-free grammars which model preferential aspects of natural language by adding probabilities to the grammar rules.

Definition 3.2 *A probabilistic context-free grammar $PCFG$ is a quintuple $\langle N, T, R, p, S \rangle$ with*

N	*finite set of non-terminal symbols*
T	*finite set of terminal symbols,* $T \cap N = \emptyset$
R	*finite set of rules* $C \rightarrow \gamma$, $C \in N$ *and* $\gamma \in (N \cup T)^*$
p	*corresponding finite set of probabilities on rules,* $(\forall r \in R) \ : \ 0 \leq p(r) \leq 1$ *and* $(\forall C \in N) \ : \ \sum_\gamma p(C \rightarrow \gamma) = 1$
S	*distinguished start symbol,* $S \in N$

The probability of a syntactic tree analysis $p(t)$ for a sentence is defined as the product of probabilities for the rules r applied in the tree. The frequency of a rule r in the respective tree is given by $f_t(r)$. On the basis of parse tree probabilities for sentences or parts of sentences, PCFGs rank syntactic analyses according to their plausibility.

$$p(t) = \prod_{r \ in \ R} p(r)^{f_t(r)} \tag{3.1}$$

As an example, consider the probabilistic context-free grammar in Table 3.2. The grammar assigns ambiguous analyses to the sentence *John ate that cake*, as in Figure 3.2. (The rule probabilities are marked as subscripts on the respective parent categories.) According to the grammar rules, the demonstrative pronoun can either represent a stand-alone noun phrase or combine with a common noun to form a noun phrase. Assuming equal probabilities of 0.5 for both verb phrase types $\langle$ V NP $\rangle$ and $\langle$ V NP NP $\rangle$ and equal probabilities of 0.3 for both noun phrase types $\langle$ N $\rangle$ and $\langle$ DEM N $\rangle$, the probabilities for the complete trees are 0.045 for the first analysis compared to 0.0045 for the second one. In this example, the probabilistic grammar resolves the structural noun phrase ambiguity in the desired way, since the probability for the preferred first (transitive) tree is larger than for the second (ditransitive) tree.

N	S, NP, PN, N, DEM, VP, V	
T	*John, cake, ate, that*	
R,p	S $\rightarrow$ NP VP,	p(S $\rightarrow$ NP VP) = 1,
	NP $\rightarrow$ PN,	p(NP $\rightarrow$ PN) = 0.3,
	NP $\rightarrow$ N,	p(NP $\rightarrow$ N) = 0.3,
	NP $\rightarrow$ DEM,	p(NP $\rightarrow$ DEM) = 0.1,
	NP $\rightarrow$ DEM N,	p(NP $\rightarrow$ DEM N) = 0.3,
	VP $\rightarrow$ V NP,	p(VP $\rightarrow$ V NP) = 0.5,
	VP $\rightarrow$ V NP NP,	p(VP $\rightarrow$ V NP NP) = 0.5,
	PN $\rightarrow$ *John*,	p(PN $\rightarrow$ *John*) = 1,
	N $\rightarrow$ *cake*,	p(N $\rightarrow$ *cake*) = 1,
	V $\rightarrow$ *ate*,	p(V $\rightarrow$ *ate*) = 1,
	DEM $\rightarrow$ *that*	p(DEM $\rightarrow$ *that*) = 1
S	S	

Table 3.2: Example PCFG (1)

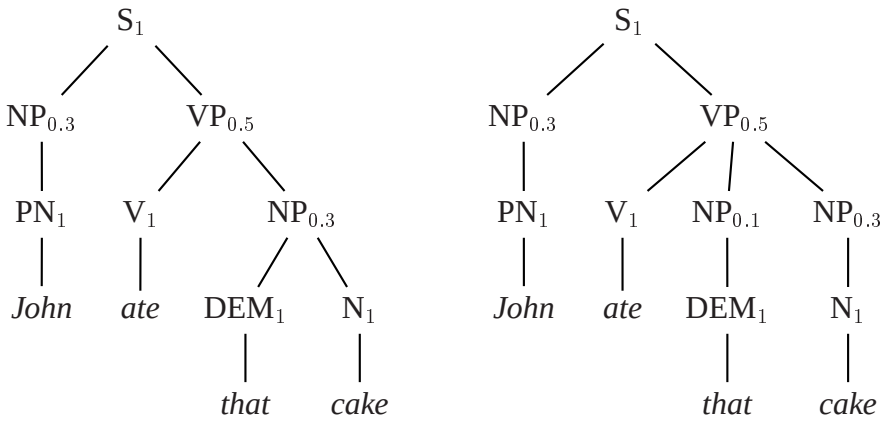


Figure 3.2: Syntactic analyses for *John ate that cake*

Now consider the probabilistic context-free grammar in Table 3.3. The grammar is ambiguous with respect to prepositional phrase attachment: prepositional phrases can either be attached to a noun phrase by NP $\rightarrow$ NP PP or to a verb phrase by VP $\rightarrow$ VP PP. The grammar assigns ambiguous analyses to the sentence *John eats the cake with a spoon*[2] as illustrated in Figure 3.3.

N	S, NP, PN, N, VP, V, PP, P, DET	
T	*John, cake, icing, spoon, eats, the, a, with*	
R,p	S $\rightarrow$ NP VP,	p(S $\rightarrow$ NP VP) = 1,
	NP $\rightarrow$ PN,	p(NP $\rightarrow$ PN) = 0.3,
	NP $\rightarrow$ N,	p(NP $\rightarrow$ N) = 0.25,
	NP $\rightarrow$ DET N,	p(NP $\rightarrow$ DET N) = 0.25,
	NP $\rightarrow$ NP PP,	p(NP $\rightarrow$ NP PP) = 0.2,
	VP $\rightarrow$ V NP,	p(VP $\rightarrow$ V NP) = 0.7,
	VP $\rightarrow$ VP PP,	p(VP $\rightarrow$ VP PP) = 0.3,
	PP $\rightarrow$ P NP,	p(PP $\rightarrow$ P NP) = 1,
	PN $\rightarrow$ *John*,	p(PN $\rightarrow$ *John*) = 1,
	N $\rightarrow$ *cake*,	p(N $\rightarrow$ *cake*) = 0.4,
	N $\rightarrow$ *icing*,	p(N $\rightarrow$ *icing*) = 0.3,
	N $\rightarrow$ *spoon*,	p(N $\rightarrow$ *spoon*) = 0.3,
	V $\rightarrow$ *eats*,	p(V $\rightarrow$ *eats*) = 1,
	P $\rightarrow$ *with*,	p(P $\rightarrow$ *with*) = 1,
	DET $\rightarrow$ *the*,	p(DET $\rightarrow$ *the*) = 0.5,
	DET $\rightarrow$ *a*	p(DET $\rightarrow$ *a*) = 0.5
S	S	

Table 3.3: Example PCFG (2)

The analyses show a preference for correctly attaching the prepositional phrase *with a spoon* as instrumental modifier to the verb phrase instead of the noun phrase: the probability of the former parse tree is $2.36 * 10^{-4}$ compared to the probability of the latter parse tree $1.58 * 10^{-4}$. This preference is based on the rule probabilities in the grammar which prefer verb phrase attachment (0.3) over noun phrase attachment (0.2).

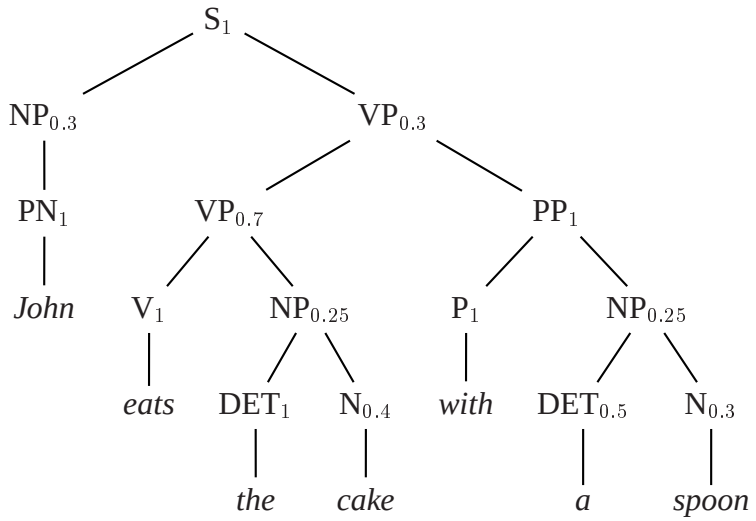
The same grammar assigns ambiguous analyses to the sentence *John eats the cake with icing* as in Figure 3.4. In this case, the preferred attachment of the prepositional phrase *with icing* would be as modifier of the noun phrase *the cake*, but the grammar assigns a probability of $3.15 * 10^{-4}$ to the noun phrase attachment (first analysis) compared to a probability of $4.73 * 10^{-4}$ for the attachment to the verb phrase (second analysis). As in the preceding example, the structural preference for the verb phrase attachment over the noun phrase attachment is based on the attachment probabilities in the grammar.

[2]The two example sentences in Figures 3.3 and 3.4 are taken from Manning and Schütze (1999, page 278).

The examples illustrate that probabilistic context-free grammars realise PP-attachment structurally, without considering the lexical context. PCFGs assign preferences to structural units on basis of grammar rule probabilities, but they do not distinguish rule applications with reference to the lexical heads of the rules. With respect to the examples, they either have a preference for PP-attachment to the verb or to the noun, but they do not recognise that *spoon* is an instrument for *to eat* or that *icing* describes the topping of the *cake*.

In addition to defining structural preferences, PCFGs can model degrees of acceptability. For example, a German grammar might define preferences on case assignment; genitive noun phrases are nowadays partly replaced by dative noun phrases: (i) A genitive noun phrase subcategorised by the preposition *wegen* 'because of' is commonly replaced by a dative noun phrase, cf. *wegen des Regens*$_{Gen}$ and *wegen dem Regen*$_{Dat}$ 'because of the rain'. (ii) Genitive noun phrases subcategorised by the verb *gedenken* 'commemorate' are often replaced by dative noun phrases, cf. *der Menschen*$_{Gen}$ *gedenken* and *den Menschen*$_{Dat}$ *gedenken* 'commemorate the people', but the substitution is less common than in (i). (iii) Genitive noun phrases modifying common nouns cannot be replaced by dative noun phrases, cf. *der Hut des Mannes*$_{Gen}$ and **der Hut dem Mann*$_{Dat}$ 'the hat of the man'. Concluding the examples, PCFGs can define degrees of case acceptability for noun phrases depending on their structural embedding.

To summarise, PCFGs are an extension of context-free grammars in that they can model structural preferences (as for noun phrase structure), and degrees of acceptability (such as case assignment). But PCFGs fail when it comes to lexically sensitive phenomena such as PP-attachment, or selectional preferences of individual verbs, since they are based purely on structural factors.



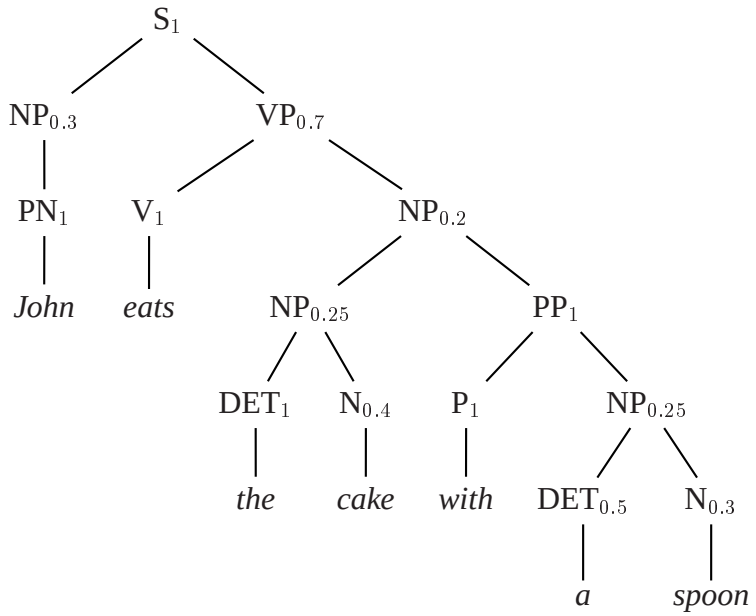


Figure 3.3: Syntactic analyses for *John eats the cake with a spoon*

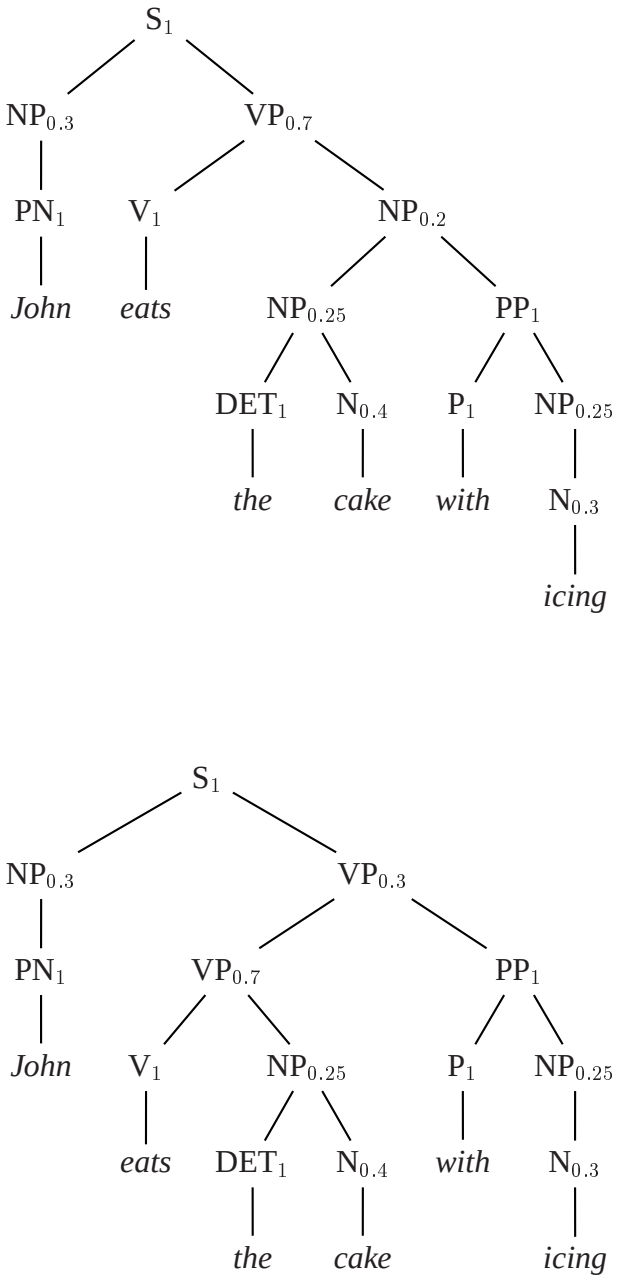


Figure 3.4: Syntactic analyses for *John eats the cake with icing*

3.1.3 Head-Lexicalised Probabilistic Context-Free Grammars

Various extensions of PCFGs are possible. Since the main drawback of PCFGs concerns their inability of modelling lexical dependencies, a common idea behind PCFG extensions is their expansion with lexical information. Examples are the decision trees in Magerman (1995), parsing models in Collins (1997), bilexical grammars in Eisner and Satta (1999), and maximum entropy modelling in Charniak (2000).

The approach as used in this thesis defines head-lexicalised probabilistic context-free grammars (H-L PCFGs) as a lexicalised extension of PCFGs. The idea of the grammar model originates from Charniak (1995) and has been implemented at the IMS Stuttgart by Carroll (1997) to learn valencies for English verbs (Carroll and Rooth, 1998). This work uses a re-implementation by Schmid (2000). Like other approaches, H-L PCFGs extend the idea of PCFGs by incorporating the lexical head of each rule into the grammar parameters. The lexical incorporation is realised by marking the head category on the right hand side of each context-free grammar rule, e.g. VP $\rightarrow$ V' NP. Each category in the rule bears a lexical head, and the lexical head from the head child category is propagated to the parent category. The lexical head of a terminal category is the respective full or lemmatised word form.

The lexical head marking in the grammar rules enables the H-L PCFG to instantiate the following grammar parameters, as defined by Schmid (2000):

- $p_{start}(s)$ is the probability that s is the category of the root node of a parse tree.
- $p_{start}(h|s)$ is the probability that a root node of category s bears the lexical head h.
- $p_{rule}(r|C,h)$ is the probability that a (parent) node of category C with lexical head h is expanded by the grammar rule r.
- $p_{choice}(h_C|C_P,h_P,C_C)$ is the probability that a (non-head) child node of category C_C bears the lexical head h_C, the parent category is C_P and the parent head is h_P.

In case a H-L PCFG does not include lemmatisation of its terminal symbols, either the lexical head h of a terminal node and the full word form $w \in T$ are identical and $p_{rule}(C \rightarrow w|C,h)$ is 1 (e.g. $p_{rule}(C \rightarrow runs|C,runs) = 1$), or the lexical head differs from the word form and $p_{rule}(C \rightarrow w|C,h)$ is 0 (e.g. $p_{rule}(C \rightarrow runs|C,ran) = 0$). In case a grammar does include lemmatisation of its terminal symbols, the probability $p_{rule}(C \rightarrow w|C,h)$ is distributed over the different word forms w with common lemmatised lexical head h (e.g. $p_{rule}(C \rightarrow runs|C,run) = 0.3$, $p_{rule}(C \rightarrow run|C,run) = 0.2$, $p_{rule}(C \rightarrow ran|C,run) = 0.5$).

The probability of a syntactic tree analysis $p(t)$ for a sentence is defined as the product of the probabilities for the start category s, the rules r, and the relevant lexical heads h which are included in the tree, cf. Equation 3.2. R refers to the set of rules established by the grammar, N to the set of non-terminal categories, and T to the set of terminal categories. Frequencies in the tree analysis are referred to by $f_t(r,C,h)$ for lexical rule parameters and $f_t(h_C,C_P,h_P,C_C)$

for lexical choice parameters. H-L PCFGs are able to rank syntactic analyses including lexical choices.

$$\begin{aligned} p(t) = & \; p_{start}(s) \; * \\ & \; p_{start}(h|s) \; * \\ & \prod_{r \in R, C \in N, h \in T} p_{rule}(r|C,h)^{f_t(r,C,h)} \; * \\ & \prod_{C_P, C_C \in N; h_P, h_C \in T} p_{choice}(h_C|C_P, h_P, C_C)^{f_t(h_C, C_P, h_P, C_C)} \end{aligned} \tag{3.2}$$

As example, consider the head-lexicalised probabilistic context-free grammar in Tables 3.4 and 3.5. Table 3.4 defines the grammar rules, with the heads of the rules marked by an apostrophe. The probability distributions on the lexicalised grammar parameters are given in Table 3.5. To distinguish terminal symbols and lexical heads (here: lemmatised word forms), the terminal symbols are printed in *italic* letters, the lexical heads in `typewriter` font.

N	S, NP, PN, N, VP, V, PP, P, POSS
T	*John, Mary, anger, smile, blames, loves, for, her*
R	S $\rightarrow$ NP VP',
	NP $\rightarrow$ PN', NP $\rightarrow$ POSS N',
	VP $\rightarrow$ VP' PP, VP $\rightarrow$ V' NP, VP $\rightarrow$ V' NP PP,
	PP $\rightarrow$ P' NP,
	PN $\rightarrow$ *John*', PN $\rightarrow$ *Mary*',
	N $\rightarrow$ *anger*', N $\rightarrow$ *smile*',
	V $\rightarrow$ *blames*', V $\rightarrow$ *loves*',
	P $\rightarrow$ *for*', POSS $\rightarrow$ *her*'
S	S

Table 3.4: Example H-L PCFG (rules)

According to the maximum probability parse, the H-L PCFG analyses the sentence *John blames Mary for her anger* as in Figure 3.5, with the prepositional phrase *for her anger* correctly analysed as argument of the verb. The sentence *John loves Mary for her smile* is analysed as in Figure 3.6, with the prepositional phrase *for her smile* correctly analysed as adjunct to the verb phrase. In the trees, the lexical heads of the grammar categories are cited as superscripts of the categories. p_{start} is quoted on the left of the root nodes S. For each node in the tree, p_{rule} is quoted on the right of the category, and p_{choice} is quoted on the right of each child category.

Multiplying the probabilities in the trees results in a probability of $8.7 * 10^{-3}$ for *John blames Mary for her anger* in Figure 3.5 and a probability of $1.9 * 10^{-3}$ for *John loves Mary for her smile* in Figure 3.6. If the *blame* sentence had been analysed incorrectly with the prepositional phrase *for her anger* as adjunct to the verb phrase, or the *love* sentence with the prepositional phrase *for her smile* as argument of the verb, the probabilities would have been $4.3 * 10^{-4}$ and $1.1 * 10^{-3}$

p_{start}	p_{start}(S) = 1,	
	p_{start}(`blame`|S) = 0.5,	p_{start}(`love`|S) = 0.5
p_{rule}	p_{rule}(S → NP VP' | S, `blame`) = 1,	p_{rule}(S → NP VP' | S, `love`) = 1,
	p_{rule}(NP → PN' | NP, `John`) = 0.9,	p_{rule}(NP → POSS N' | NP, `John`) = 0.1,
	p_{rule}(NP → PN' | NP, `Mary`) = 0.9,	p_{rule}(NP → POSS N' | NP, `Mary`) = 0.1,
	p_{rule}(NP → PN' | NP, `anger`) = 0.1,	p_{rule}(NP → POSS N' | NP, `anger`) = 0.9,
	p_{rule}(NP → PN' | NP, `smile`) = 0.1,	p_{rule}(NP → POSS N' | NP, `smile`) = 0.9,
	p_{rule}(VP → VP' PP | VP, `blame`) = 0.1,	p_{rule}(VP → VP' PP | VP, `love`) = 0.3,
	p_{rule}(VP → V' NP | VP, `blame`) = 0.3,	p_{rule}(VP → V' NP | VP, `love`) = 0.6,
	p_{rule}(VP → V' NP PP | VP, `blame`) = 0.6,	p_{rule}(VP → V' NP PP | VP, `love`) = 0.1,
	p_{rule}(PN → *John*' | PN, `John`) = 1	p_{rule}(PN → *Mary*' | PN, `John`) = 0,
	p_{rule}(PN → *Mary*' | PN, `Mary`) = 1	p_{rule}(PN → *John*' | PN, `Mary`) = 0,
	p_{rule}(N → *anger*' | N, `anger`) = 1,	p_{rule}(N → *smile*' | N, `anger`) = 0,
	p_{rule}(N → *smile*' | N, `smile`) = 1,	p_{rule}(N → *anger*' | N, `smile`) = 0,
	p_{rule}(V → *blames*' | V, `blame`) = 1,	p_{rule}(V → *loves*' | V, `blame`) = 0,
	p_{rule}(V → *loves*' | V, `love`) = 1,	p_{rule}(V → *blames*' | V, `love`) = 0,
	p_{rule}(PP → P' NP | PP, `for`) = 1,	p_{rule}(P → *for*' | P, `for`) = 1,
	p_{rule}(POSS → *her*' | POSS, `she`) = 1	
p_{choice}	p_{choice}(`John` | S, `blame`, NP) = 0.4,	p_{choice}(`Mary` | S, `blame`, NP) = 0.4,
	p_{choice}(`anger` | S, `blame`, NP) = 0.1,	p_{choice}(`smile` | S, `blame`, NP) = 0.1,
	p_{choice}(`John` | S, `love`, NP) = 0.4,	p_{choice}(`Mary` | S, `love`, NP) = 0.4,
	p_{choice}(`anger` | S, `love`, NP) = 0.1,	p_{choice}(`smile` | S, `love`, NP) = 0.1,
	p_{choice}(`she` | NP, `John`, POSS) = 1,	p_{choice}(`she` | NP, `anger`, POSS) = 1,
	p_{choice}(`she` | NP, `Mary`, POSS) = 1,	p_{choice}(`she` | NP, `smile`, POSS) = 1,
	p_{choice}(`for` | VP, `blame`, PP) = 1,	p_{choice}(`for` | VP, `love`, PP) = 1,
	p_{choice}(`John` | VP, `blame`, NP) = 0.4,	p_{choice}(`Mary` | VP, `blame`, NP) = 0.4,
	p_{choice}(`anger` | VP, `blame`, NP) = 0.1,	p_{choice}(`smile` | VP, `blame`, NP) = 0.1,
	p_{choice}(`John` | VP, `love`, NP) = 0.3,	p_{choice}(`Mary` | VP, `love`, NP) = 0.3,
	p_{choice}(`anger` | VP, `love`, NP) = 0.2,	p_{choice}(`smile` | VP, `love`, NP) = 0.2,
	p_{choice}(`John` | PP, `for`, NP) = 0.25,	p_{choice}(`Mary` | PP, `for`, NP) = 0.25,
	p_{choice}(`anger` | PP, `for`, NP) = 0.25,	p_{choice}(`smile` | PP, `for`, NP) = 0.25

Table 3.5: Example H-L PCFG (lexicalised parameters)

respectively, i.e. the correct analyses of the sentences in Figures 3.5 and 3.6 are more probable than their incorrect counterparts. This distinction in probabilities results from the grammar parameters which reflect the lexical preferences of the verbs, in this example concerning their subcategorisation properties. For *blame*, subcategorising the transitive ⟨ V NP PP ⟩ including the PP is more probable than subcategorising the intransitive ⟨ V NP ⟩ , and for *love* the lexical preference is vice versa.

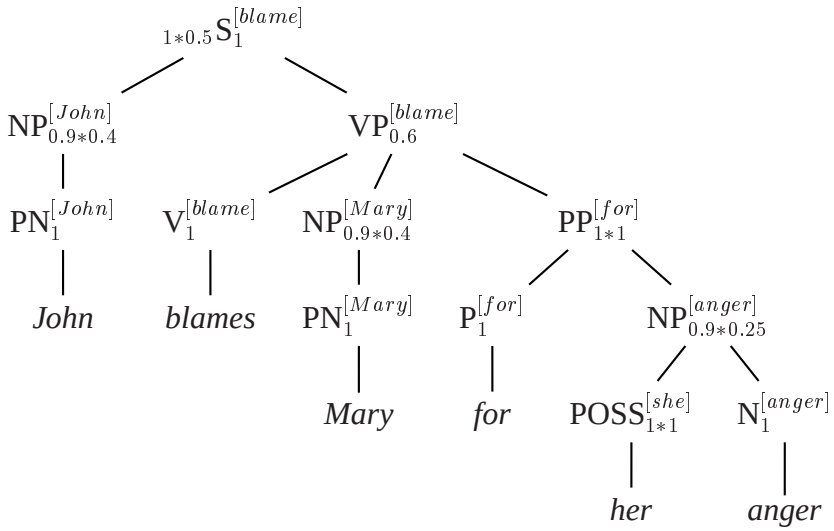


Figure 3.5: Syntactic analysis for *John blames Mary for her anger*

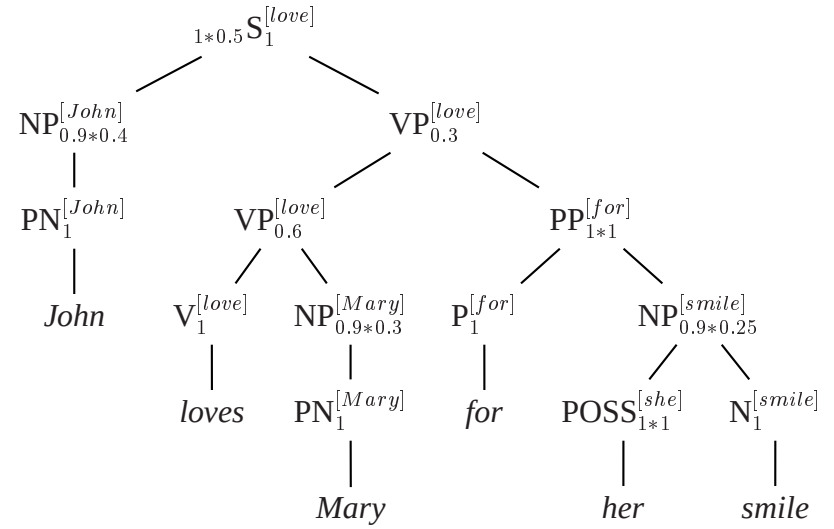


Figure 3.6: Syntactic analysis for *John loves Mary for her smile*

To summarise, H-L PCFGs are a further extension of context-free grammars in that they can model structural preferences including lexical selection, such as PP-attachment and selectional argument preferences of individual verbs. According to Manning and Schütze (1999), main problems of H-L PCFGs concern (i) the assumption of context-freeness, i.e. that a certain subtree in a sentence analysis is analysed in the same way no matter where in the sentence parse it is situated; for example, noun phrase formation actually differs according to the position, since noun phrases tend to be pronouns more often in sentence initial position than elsewhere. And (ii) for discriminating the large number of parameters in a H-L PCFG, a sufficient amount of linguistic data is required. The detailed linguistic information in the grammar model is of large value, but effective smoothing techniques are necessary to overcome the sparse data problem.

3.1.4 Summary

This section has introduced the theoretical background of context-free grammars and their statistical extensions. Context-free grammars (CFGs) can model a large part of natural language structure, but fail to express preferences. Probabilistic context-free grammars (PCFGs) are an extension of context-free grammars which can model structural preferences (as for noun phrase structure) and degrees of acceptability (such as case assignment), but they fail when it comes to lexically sensitive phenomena. Head-lexicalised probabilistic context-free grammars (H-L PCFGs) are a further extension of context-free grammars in that they can model structural preferences including lexical selection, such as PP-attachment and selectional argument preferences of individual verbs.

My statistical grammar model is based on the framework of H-L PCFGs. The development of the grammar model is organised in three steps, according to the theoretical grammar levels.

1. Manual definition of CFG rules with head-specification,
2. Assigning probabilities to CFG rules (extension of CFG to PCFG),
3. Lexicalisation of the PCFG (creation of H-L PCFG).

The following Section 3.2 describes the manual definition of the CFG rules (step 1) in detail, and Section 3.3 describes the grammar extension and training with respect to steps 2 and 3.

3.2 Grammar Development and Implementation

This section describes the development and implementation of the German context-free grammar. As explained above, the context-free backbone is the basis for the lexicalised probabilistic extension which is used for learning the statistical grammar model. Section 3.2.1 introduces the specific aspects of the grammar development which are important for the acquisition of lexicon-relevant verb information. Section 3.2.2 then describes the German context-free grammar rules.

3.2.1 Grammar Development for Lexical Verb Information

The context-free grammar framework is developed with regard to the overall goal of obtaining reliable lexical information on verbs. This goal influences the development process in the following ways:

- To provide a sufficient amount of training data for the model parameters, the grammar model should be robust, since the grammar needs to cover as much training data as possible. The robustness is important (i) to obtain lexical verb information for a large sample of German verbs, and (ii) to learn the grammar parameters to a reliable degree. To give an example, (i) in contrast to a former version of the German grammar by Beil *et al.* (1999) where only verb final clauses are regarded, the grammar covers all German sentence types in order to obtain as much information from the training corpus as possible. (ii) For fine-tuning the grammar parameters with regard to reliable verb subcategorisation, no restriction on word order is implemented, but all possible scrambling orders of German clauses are considered.

- Infrequent linguistic phenomena are disregarded if they are likely to confuse the learning of frequent phenomena. For example, coherent clauses might be structurally merged, such that it is difficult to distinguish main and subcategorised clause without crossing edges. Example (3.3) shows a merging of a non-finite and a relative clause. *sie* is the subject of the control verb *versprochen* and also embedded in the non-finite clause *den zu lieben* subcategorised by *versprochen*. Implementing the phenomenon in the grammar would enable us to parse such sentences, but at the same time include an enormous source for ambiguities and errors in the relatively free word order language German, so the implementation is ignored. The mass of training data is supposed to compromise for the parsing failure of infrequent phenomena.

  (3.3) *den sie zu lieben versprochen hat*
  whom she to love promised has
  'whom she has promised to love'

- Work effort concentrates on defining linguistic structures which are relevant to lexical verb information, especially subcategorisation. On the one hand, this results in fine-grained structural levels for subcategorisation. For example, for each clause type I define an extraordinary rule level
  `C-<type>` $\rightarrow$ `S-<type>.<frame>`
  where the clause level `C` produces the clause category `S` which is accompanied by the subcategorisation frame for the clause. A lexicalisation of the grammar rules with their verb heads automatically leads to a distribution over frame types. In addition, the parsing strategy is organised in an exceptional way: Since the lexical verb head as the bearer of the clausal subcategorisation needs to be propagated through the parse tree, the grammar structures are based on a so-called 'collecting strategy' around the verb head, no matter in

which topological position the verb head is or whether the verb head is realised as a finite or non-finite verb.
On the other hand, structural levels for constituents outside verb subcategorisation are ignored. For example, adjectival and adverbial phrases are realised by simple lists, which recognise the phrases reliably, but disregard fine-tuning of their internal structure.

- The grammar framework needs to control the number of parameters, especially when it comes to the lexicalised probabilistic extension of the context-free grammar. This is realised by keeping the category features in the grammar to a minimum. For example, the majority of noun phrases is recognised reliably with the case feature only, disregarding number and gender. The latter features are therefore disregarded in the context-free grammar.

The above examples concerning the grammar development strategy illustrate that the context-free grammar defines linguistic structures in an unusual way. This is so because the main goal of the grammar is the reliable definition of lexical verb information, and we need as much information on this aspect as possible to overcome the problem of data sparseness.

3.2.2 The German Context-Free Grammar

The German context-free grammar rules are manually written. The manual definition is supported by the grammar development environment of YAP (Schmid, 1999), a feature based parsing framework, which helps the grammar developer with managing rules and features. In addition, the statistical parser LoPar (Schmid, 2000) provides a graphical interface to control the grammar development. Following, I describe the grammar implementation, starting with the grammar terminals and then focusing on the grammar rules.

Grammar Terminals

The German grammar uses morpho-syntactic terminal categories as based on the dictionary database IMSLex and the morphological analyser AMOR (Lezius *et al.*, 1999, 2000): Each word form is assigned one or multiple part-of-speech tags and the corresponding lemmas. I have adopted the morphological tagging system with task-specific changes, for example ignoring the features *gender* and *number* on verbs, nouns and adjectives. Table 3.6 gives an overview of the terminal categories to which the AMOR tags are mapped as basis for the grammar rules, Table 3.7 lists the relevant feature values, and Table 3.8 gives examples for tag-feature combinations.

Terminal Category		Features	Tag Example
attributive adjective	ADJ	case	ADJ.Akk
indeclinable adjective	ADJ-invar		ADJ-invar
predicative adjective	ADJ-pred		ADJ-pred
adverb	ADV		ADV
article	ART	case	ART.Dat
cardinal number	CARD		CARD
year number	CARD-time		CARD-time
demonstrative pronoun	DEM	distribution, case	DEM.subst.Nom
expletive pronoun	ES		ES
indefinite pronoun	INDEF	distribution, case	INDEF.attr.Dat
interjection	INTJ		INTJ
conjunction	KONJ	conjunction type	KONJ.Sub
proper name	NE	case	NE.Nom
common noun	NN	case	NN.Gen
ordinal number	ORD		ORD
possessive pronoun	POSS	distribution, case	POSS.attr.Akk
postposition	POSTP	case, postposition	POSTP.Dat.entlang
reflexive pronoun	PPRF	case	PPRF.Dat
personal pronoun	PPRO	case	PPRO.Nom
reciprocal pronoun	PPRZ	case	PPRZ.Akk
preposition	PREP	case, preposition	PREP.Akk.ohne
preposition + article	PREPart	case, preposition	PREPart.Dat.zu
pronominal adverb	PROADV	pronominal adverb	PROADV.dazu
particle	PTKL	particle type	PTKL.Neg
relative pronoun	REL	distribution, case	REL.subst.Nom
sentence symbol	S-SYMBOL	symbol type	S-SYMBOL.Komma
truncated word form	TRUNC		TRUNC
finite verb	V*X*FIN *X* = { B(leiben), H(aben), M(odal), S(ein), V(oll), W(erden) }		VMFIN
finite verb (part of separable verb)	VVFINsep		VVFINsep
infinitival verb	V*X*INF *X* = { B(leiben), H(aben), M(odal), S(ein), V(oll), W(erden) }		VWINF
infinitival verb (incorporating *zu*)	VVIZU		VVIZU
past participle	V*X*past *X* = { B(leiben), M(odal), S(ein), V(oll), W(erden) }		VVpast
verb prefix	VPRE		VPRE
interrogative adverb	WADV	interrogative adverb	WADV.wann
interrogative pronoun	WPRO	distribution, case	WPRO.attr.Gen

Table 3.6: Terminal grammar categories

Feature	Feature Values
case	Nom, Akk, Dat, Gen
distribution	attr, subst
symbol type	Komma, Norm
conjunction type	Inf, Kon, Sub, Vgl, dass, ob
particle type	Adj, Ant, Neg, zu
preposition	[Akk] ab, an, auf, außer, bis, durch, entlang, für, gegen, gen, hinter, in, je, kontra, neben, ohne, per, pro, um, unter, versus, via, vor, wider, zwischen, über
	[Dat] ab, an, anstatt, auf, aus, außer, außerhalb, bei, binnen, dank, einschließlich, entgegen, entlang, entsprechend, exklusive, fern, gegenüber, gemäß, gleich, hinter, in, inklusive, innerhalb, laut, längs, mangels, mit, mitsamt, mittels, nach, nah, nahe, neben, nebst, nächst, samt, seit, statt, trotz, unter, von, vor, wegen, während, zu, zunächst, zwischen, ähnlich, über
	[Gen] abseits, abzüglich, anfangs, angesichts, anhand, anläßlich, anstatt, anstelle, aufgrund, ausschließlich, außer, außerhalb, beiderseits, beidseits, bezüglich, binnen, dank, diesseits, eingangs, eingedenk, einschließlich, entlang, exklusive, fern, hinsichtlich, infolge, inklusive, inmitten, innerhalb, jenseits, kraft, laut, links, längs, längsseits, mangels, minus, mittels, nahe, namens, nordwestlich, nordöstlich, nördlich,ob, oberhalb, rechts, seiten, seitens, seitlich, statt, südlich, südwestlich, südöstlich,trotz, um, unbeschadet, unerachtet, ungeachtet, unterhalb, unweit, vermittels,vermöge, orbehaltlich, wegen, westlich, während, zeit, zufolge, zugunsten, zuungunsten,zuzüglich, zwecks, östlich
postposition	[Akk] entlang, exklusive, hindurch, inklusive
	[Dat] entgegen, entlang, entsprechend, gegenüber, gemäß, nach, zufolge, zugunsten, zuliebe, zunächst, zuungunsten, zuwider
	[Gen] halber, ungeachtet, wegen, willen
pronominal adverb	dabei, dadurch, dafür, dagegen, daher, dahin, dahinter, damit, danach, daneben, daran, darauf, daraufhin, daraus, darin, darum, darunter, darüber, davon, davor, dazu, dazwischen, dementsprechend, demgegenüber, demgemäß, demnach, demzufolge, deshalb, dessenungeachtet, deswegen, dran, drauf, draus, drin, drum, drunter, drüber, hieran, hierauf, hieraufhin, hieraus, hierbei, hierdurch, hierfür, hiergegen, hierher, hierhin, hierin, hiermit, hiernach, hierum, hierunter, hiervon, hiervor, hierzu, hierüber, seitdem, trotzdem, währenddessen
interrogative adverb	wann, warum, weshalb, weswegen, wie, wieso, wieviel, wieweit, wo, wobei, wodurch, wofür, wogegen, woher, wohin, wohinein, wohinter, womit, wonach, woran, worauf, woraufhin, woraus, worein, worin, worum, worunter, worüber, wovon, wovor, wozu

Table 3.7: Terminal features

Terminal Category	Examples
ADJ.Akk	*kleine, riesiges, schönen*
ADJ-invar	*lila, mini, relaxed*
ADJ-pred	*abbruchreif, dauerhaft, schlau*
ADV	*abends, fast, immer, ratenweise*
ART.Gen	*des, einer, eines*
CARD	*0,080 5,8,14/91 dreizehn 28*
CARD-time	*1543 1920 2021*
DEM.attr.Dat / DEM.subst.Nom	*denselben, dieser, jenem* / *dasjenige, dieselben, selbige*
ES	*es*
INDEF.attr.Gen / INDEF.subst.Akk	*irgendwelcher, mehrerer* / *ebensoviele, irgendeinen, manches*
INTJ	*aha, hurra, oh, prost*
KONJ.Inf / KONJ.Kon KONJ.Sub / KONJ.Vgl KONJ.dass / KONJ.ob	*anstatt, um, ohne* / *doch, oder, und* *dass, sooft, weil* / *als, wie* *dass* / *ob*
NE.Nom	*Afrika, DDR, Julia*
NN.Dat	*ARD, C-Jugend, Häusern*
ORD	*3. 2704361.*
POSS.attr.Nom / POSS.subst.Dat	*ihr, meine, unsere* / *eurem, unseren*
POSTP.Dat.entsprechend	*entsprechend*
PPRF.Akk	*sich, uns*
PPRO.Nom	*du, ich, ihr*
PPRZ.Akk	*einander*
PREP.Akk.für	*für*
PREPart.Dat.zu	*zum*
PROADV.dadurch	*dadurch*
PTKL.Adj / PTKL.Ant PTKL.Neg / PTKL.zu	*allzu, am* / *bitte, nein* *nicht* / *zu*
REL.attr.Gen / REL.subst.Nom	*deren, dessen* / *das, der, die*
S-SYMBOL.Komma S-SYMBOL.Norm	*,* *! . : ; ?*
TRUNC	*ARD- Doktoranden- Jugend-*
VBFIN VHFIN VMFIN VSFIN VVFIN VWFIN	*bleibe, blieben* *hast, hatte* *dürftest, könnte, möchten* *sind, war, wären* *backte, ranntet, schläft* *werden, wird, würde*
VVFINsep	*gibt, rennen, trennte*
VVINF	*abblocken, eilen, schwimmen*
VVIZU	*dabeizusein, glattzubügeln*
VBpast	*geblieben*
VPRE	*ab, her, hinein, zu*
WADV.warum	*warum*
WPRO.attr.Akk / WPRO.subst.Dat	*welche, welches* / *welchen, wem*

Table 3.8: Examples of grammar terminals

Grammar Rules

The following paragraphs provide an overview of the German context-free grammar rules. Preferably the grammar code is omitted, and the rules are illustrated by syntactic trees and example sentences. Features which are irrelevant for the illustration of specific grammar rules may be left out. Explanations should help to grasp the intuition behind the rule coding strategies, cf. Section 3.2.1. The total number of context-free grammar rules is 35,821.

Sentence Structure The grammar distinguishes six finite clause types:

- `C-1-2` for verb first and verb second clauses,
- `C-rel` for relative clauses,
- `C-sub` for non-subcategorised subordinated clauses,
- `C-dass` for subcategorised subordinated *dass*-clauses ('that'-clauses),
- `C-ob` for subcategorised subordinated *ob*-clauses ('whether'-clauses),
- `C-w` for subcategorised indirect *wh*-questions.

The clause types differ with respect to their word order and their function. `C-1-2` clauses have the main verb in the first or second position of the clause, and all other claus types have the main verb in clause final position. The final clause types are distinguished, because `C-dass, C-ob` and `C-w` can represent arguments which are subcategorised by the verb, but `C-rel` and `C-sub` cannot. In addition, `C-rel` and `C-sub` have different distributions (i.e. `C-rel` typically modifies a nominal category, `C-sub` a clause), and the possible clausal arguments `C-dass, C-ob, C-w` and also `C-1-2` may be subcategorised by different verbs and verb classes.

The clause level `C` produces another the clause category `S` which is accompanied by the relevant subcategorisation frame type dominating the clause. As said before, this extraordinary rule level is provided, since the lexicalisation of the grammar rules with their verb heads will automatically lead to a distribution over frame types. The effect of this set of grammar rules will be illustrated in detail in Section 3.4 which describes the empirical lexical acquisition as based on the grammar.

```
C-<type>  ->  S-<type>.<frame>
```

In order to capture a wide range of corpus data, all possibly non-subcategorised clause types (verb first and verb second clauses, relative clauses, and non-subcategorised subordinated clauses) generate `S-top` and can be combined freely by commas and coordinating conjunctions.

```
S-top -> S-top KONJ.Kon       S-top
S-top -> S-top S-SYMBOL.Komma S-top
```

`S-top` are terminated by a full stop, question mark, exclamation mark, colon, or semicolon. `TOP` is the overall top grammar category.

```
TOP -> S-top S-SYMBOL.Norm
```

Figure 3.7 illustrates the top-level clause structure by combining a matrix clause and a non-subcategorised causal clause. The example sentence is *Peter kommt zu spät, weil er verschlafen hat* 'Peter is late, because he overslept'.

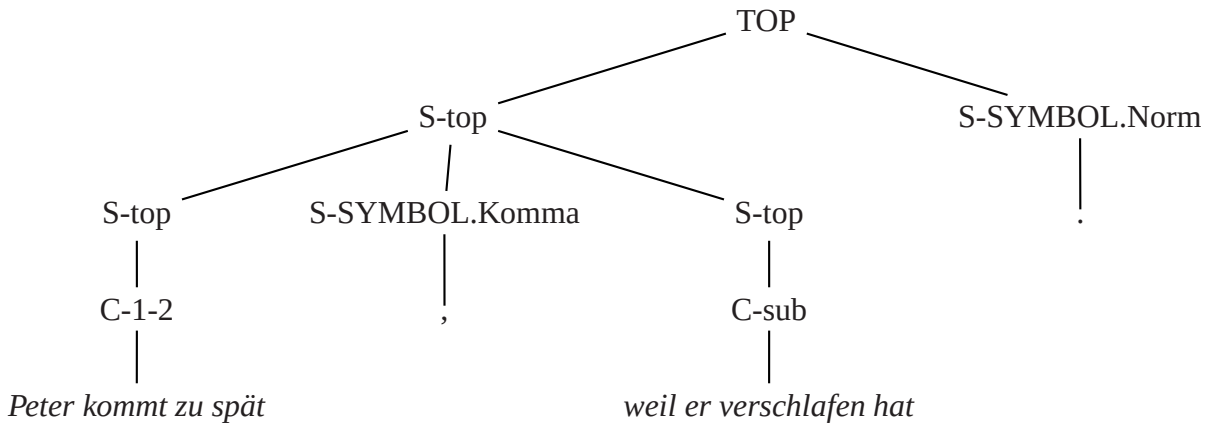


Figure 3.7: Top-level clause construction

Verb Phrases The clausal categories `S-<type>.<frame>` below `C` are generated by verb phrases which determine the clause type and the frame type. The verb phrases are the core part of the German grammar and therefore designed with special care and attention to detail. A verb phrase is defined as the verb complex which collects preceding and following arguments and adjuncts until the sentence is parsed. The resulting `S`-frame distinguishes verb arguments and verb adjuncts; it indicates the number and types of the verb arguments, verb adjuncts are not marked.

Four types of verb phrases are distinguished: active (`VPA`), passive (`VPP`), non-finite (`VPI`) verb phrases, and copula constructions (`VPK`). Each verb phrase type is accompanied by the frame type which may have maximally three arguments. Any verb can principally occur with any frame type. Possible arguments in the frames are nominative (n), dative (d) and accusative (a) noun phrases, reflexive pronouns (r), prepositional phrases (p), non-finite verb phrases (i), expletive *es* (x), and subordinated finite clauses (s-2 for verb second clauses, s-dass for *dass*-clauses, s-ob for *ob*-clauses, s-w for indirect *wh*-questions). Prepositional phrases in `VPP`, which are headed by the prepositions *von* or *durch* and indicate the deep structure subject in passive constructions, are marked by 'P' instead of 'p'. The frame types indicate the number and kind of subcategorised arguments, but they generalise over the argument order. For example, the verb phrase `VPA.nad` describes an active ditransitive verb phrase with a nominative, an accusative and a dative noun phrase (with any scrambling order); `VPA.ndp` describes an active verb phrase with a nominative and a dative noun phrase plus a prepositional phrase (with any scrambling order); `VPP.nP`

describes a passive verb phrase with a nominative noun phrase and a prepositional phrase headed by *von* or *durch* (with any scrambling order).

The combinations of verb phrases and frame types are listed in Tables 3.9 to 3.12; the active frame types in Table 3.9 generalise over the subcategorisation behaviour of the verbs[3] and have already been introduced in Appendix A. The frame types are developed with reference to the standard German grammar by Helbig and Buscha (1998). The total of 38 frame types covers the vast majority of the verb structures, only few infrequent frame types such as `naa` or `nag` have been ignored.

Active and passive verb phrases are abstracted from their voice by introducing a generalising level. For example, the clause category `S.na`, a transitive type subcategorising a direct object, produces `VPA.na` in active voice and `VPP.n` and `VPP.nP` in passive voice. This treatment is justified by argument agreement of the frame types on the deep structure level, e.g. the surface structure subject in `VPP.n` and `VPP.nP` agrees with the surface structure object in `VPA.na`, and the prepositional phrase in `VPP.nP` agrees with the surface structure subject in `VPA.na`. With 'agreement' I refer to the selectional preferences of the verbs with respect to a frame type and the frame arguments. In addition to generalising over voice, the different kinds of copula constructions in Table 3.12 are generalised to the frame type 'k'. The generalisation is performed for all `S`-types. Table 3.13 provides a list of all generalised frame descriptions. `VPI` do not represent finite clauses and therefore do not generate `S`, but are instead arguments within the `S` frame types.

[3]This idea will be explained in detail below.

Frame Type	Example
n	*Natalie$_n$ schwimmt.*
na	*Hans$_n$ sieht seine Freundin$_a$.*
nd	*Er$_n$ glaubt den Leuten$_d$ nicht.*
np	*Die Autofahrer$_n$ achten besonders auf Kinder$_p$.*
nad	*Anna$_n$ verspricht ihrem Vater$_d$ ein tolles Geschenk$_a$.*
nap	*Die kleine Verkäuferin$_n$ hindert den Dieb$_a$ am Stehlen$_p$.*
ndp	*Der Moderator$_n$ dankt dem Publikum$_d$ für sein Verständnis$_p$.*
ni	*Mein Freund$_n$ versucht immer wieder, pünktlich zu kommen$_i$.*
nai	*Er$_n$ hört seine Mutter$_a$ ein Lied singen$_i$.*
ndi	*Helene$_n$ verspricht ihrem Großvater$_d$ ihn bald zu besuchen$_i$.*
nr	*Die kleinen Kinder$_n$ fürchten sich$_r$.*
nar	*Der Unternehmer$_n$ erhofft sich$_r$ baldigen Aufwind$_a$.*
ndr	*Sie$_n$ schließt sich$_r$ nach 10 Jahren wieder der Kirche$_d$ an.*
npr	*Der Pastor$_n$ hat sich$_r$ als der Kirche würdig$_p$ erwiesen.*
nir	*Die alte Frau$_n$ stellt sich$_r$ vor, den Jackpot zu gewinnen$_i$.*
x	*Es$_x$ blitzt.*
xa	*Es$_x$ gibt viele Bücher$_a$.*
xd	*Es$_x$ graut mir$_d$.*
xp	*Es$_x$ geht um ein tolles Angebot für einen super Computer$_p$.*
xr	*Es$_x$ rechnet sich$_r$.*
xs-dass	*Es$_x$ heißt, dass Thomas sehr klug ist$_{s-dass}$.*
ns-2	*Der Abteilungsleiter$_n$ hat gesagt, er halte bald einen Vortrag$_{s-2}$.*
nas-2	*Der Chef$_n$ schnauzt ihn$_a$ an, er sei ein Idiot$_{s-2}$.*
nds-2	*Er$_n$ sagt seiner Freundin$_d$, sie sei zu krank zum Arbeiten$_{s-2}$.*
nrs-2	*Der traurige Vogel$_n$ wünscht sich$_r$, sie bliebe bei ihm$_{s-2}$.*
ns-dass	*Der Winter$_n$ hat schon angekündigt, dass er bald kommt$_{s-dass}$.*
nas-dass	*Der Vater$_n$ fordert seine Tochter$_a$ auf, dass sie verreist$_{s-dass}$.*
nds-dass	*Er$_n$ sagt seiner Geliebten$_d$, dass er verheiratet ist$_{s-dass}$.*
nrs-dass	*Der Junge$_n$ wünscht sich$_r$, dass seine Mutter bleibt$_{s-dass}$.*
ns-ob	*Der Chef$_n$ hat gefragt, ob die neue Angestellte den Vortrag hält$_{s-ob}$.*
nas-ob	*Anton$_n$ fragt seine Frau$_a$, ob sie ihn liebt$_{s-ob}$.*
nds-ob	*Der Nachbar$_n$ ruft der Frau$_d$ zu, ob sie verreist$_{s-ob}$.*
nrs-ob	*Der Alte$_n$ wird sich$_r$ erinnern, ob das Mädchen dort war$_{s-ob}$.*
ns-w	*Der kleine Junge$_n$ hat gefragt, wann die Tante endlich ankommt$_{s-w}$.*
nas-w	*Der Mann$_n$ fragt seine Freundin$_a$, warum sie ihn liebt$_{s-w}$.*
nds-w	*Der Vater$_n$ verrät seiner Tochter$_d$ nicht, wer zu Besuch kommt$_{s-w}$.*
nrs-w	*Das Mädchen$_n$ erinnert sich$_r$, wer zu Besuch kommt$_{s-w}$.*
k	*Der neue Nachbar$_k$ ist ein ziemlicher Idiot.*

Table 3.9: Subcategorisation frame types: VPA

Frame Type	Example
n	*Peter$_n$ wird betrogen.*
nP	*Peter$_n$ wird von seiner Freundin$_P$ betrogen.*
d	*Dem Vater$_d$ wird gehorcht.*
dP	*Dem Vater$_d$ wird von allen Kindern$_P$ gehorcht.*
p	*An die Vergangenheit$_p$ wird appelliert.*
pP	*Von den alten Leuten$_P$ wird immer an die Vergangenheit$_p$ appelliert.*
nd	*Ihm$_d$ wurde die Verantwortung$_n$ übertragen.*
ndP	*Ihm$_d$ wurde von seinem Chef$_P$ die Verantwortung$_n$ übertragen.*
np	*Anna$_n$ wurde nach ihrer Großmutter$_p$ benannt.*
npP	*Anna$_n$ wurde von ihren Eltern$_P$ nach ihrer Großmutter$_p$ benannt.*
dp	*Der Organisatorin$_d$ wird für das Essen$_p$ gedankt.*
dpP	*Der Organisatorin$_d$ wird von ihren Kollegen$_P$ für das Essen$_p$ gedankt.*
i	*Pünktlich zu gehen$_i$ wurde versprochen.*
iP	*Von den Schülern$_P$ wurde versprochen, pünktlich zu gehen$_i$.*
ni	*Der Sohn$_n$ wurde verpflichtet, seiner Mutter zu helfen$_i$.*
niP	*Der Sohn$_n$ wurde von seiner Mutter$_P$ verpflichtet, ihr zu helfen$_i$.*
di	*Dem Vater$_d$ wurde versprochen, früh ins Bett zu gehen$_i$.*
diP	*Dem Vater$_d$ wurde von seiner Freundin$_P$ versprochen, früh ins Bett zu gehen$_i$.*
s-2	*Der Chef halte einen Vortrag$_{s-2}$, wurde angekündigt.*
sP-2	*Vom Vorstand$_P$ wurde angekündigt, der Chef halte einen Vortrag$_{s-2}$.*
ns-2	*Peter$_n$ wird angeschnauzt, er sei ein Idiot$_{s-2}$.*
nsP-2	*Peter$_n$ wird von seiner Freundin$_P$ angeschnauzt, er sei ein Idiot$_{s-2}$.*
ds-2	*Dem Mädchen$_d$ wird bestätigt, sie werde reich$_{s-2}$.*
dsP-2	*Dem Mädchen$_d$ wird vom Anwalt$_P$ bestätigt, sie werde reich$_{s-2}$.*
s-dass	*Dass er den Vortrag hält$_{s-dass}$, wurde rechtzeitig angekündigt.*
sP-dass	*Dass er den Vortrag hält$_{s-dass}$, wurde rechtzeitig vom Chef$_P$ angekündigt.*
ns-dass	*Die Mutter$_n$ wurde aufgefordert, dass sie verreist$_{s-dass}$.*
nsP-dass	*Die Mutter$_n$ wurde von ihrem Freund$_P$ aufgefordert, dass sie verreist$_{s-dass}$.*
ds-dass	*Dem Mädchen$_d$ wird bestätigt, dass sie reich sein wird$_{s-dass}$.*
dsP-dass	*Dem Mädchen$_d$ wird vom Anwalt$_P$ bestätigt, dass sie reich sein wird$_{s-dass}$.*
s-ob	*Ob er den Vortrag hält$_{s-ob}$, wurde gefragt.*
sP-ob	*Ob er den Vortrag hält$_{s-ob}$, wurde vom Vorstand$_P$ gefragt.*
ns-ob	*Anna$_n$ wurde gefragt, ob sie ihren Freund liebt$_{s-ob}$.*
nsP-ob	*Anna$_n$ wurde von ihrem Freund$_P$ gefragt, ob sie ihn liebt$_{s-ob}$.*
ds-ob	*Dem Mädchen$_d$ wird bestätigt, ob sie reich sein wird$_{s-ob}$.*
dsP-ob	*Dem Mädchen$_d$ wird vom Anwalt$_P$ bestätigt, ob sie reich sein wird$_{s-ob}$.*
s-w	*Wann er den Vortrag hält$_{s-w}$, wurde gefragt.*
sP-w	*Wann er den Vortrag hält$_{s-w}$, wurde vom Vorstand$_P$ gefragt.*
ns-w	*Die Mutter$_n$ wurde gefragt, wann sie verreist$_{s-w}$.*
nsP-w	*Die Mutter$_n$ wurde von ihrem Freund$_P$ gefragt, wann sie verreist$_{s-w}$.*
ds-w	*Dem Kind$_d$ wird gesagt, wer zu Besuch kommt$_{s-w}$.*
dsP-w	*Dem Kind$_d$ wird von den Eltern$_P$ gesagt, wer zu Besuch kommt$_{s-w}$.*

Table 3.10: Subcategorisation frame types: VPP

Frame Type	Example
-	*zu schlafen*
a	*ihn$_a$ zu verteidigen*
d	*ihr$_d$ zu helfen*
p	*an die Vergangenheit$_p$ zu appellieren*
ad	*seiner Mutter$_d$ das Geschenk$_a$ zu geben*
ap	*ihren Freund$_a$ am Gehen$_p$ zu hindern*
dp	*ihr$_d$ für die Aufmerksamkeit$_p$ zu danken*
r	*sich$_r$ zu erinnern*
ar	*sich$_r$ Aufwind$_a$ zu erhoffen*
dr	*sich$_r$ der Kirche$_d$ anzuschließen*
pr	*sich$_r$ für den Frieden$_p$ einzusetzen*
s-2	*anzukündigen, er halte einen Vortrag$_{s-2}$*
as-2	*ihn$_a$ anzuschnauzen, er sei ein Idiot$_{s-2}$*
ds-2	*ihr$_d$ zu sagen, sie sei unmöglich$_{s-2}$*
s-dass	*anzukündigen, dass er einen Vortrag hält$_{s-dass}$*
as-dass	*sie$_a$ aufzufordern, dass sie verreist$_{s-dass}$*
ds-dass	*ihr$_d$ zu sagen, dass sie unmöglich sei$_{s-dass}$*
s-ob	*zu fragen, ob sie ihn liebe$_{s-ob}$*
as-ob	*sie$_a$ zu fragen, ob sie ihn liebe$_{s-ob}$*
ds-ob	*ihr$_d$ zuzurufen, ob sie verreist$_{s-ob}$*
s-w	*zu fragen, wer zu Besuch kommt$_{s-w}$*
as-w	*sie$_a$ zu fragen, wer zu Besuch kommt$_{s-w}$*
ds-w	*ihr$_d$ zu sagen, wann der Besuch kommt$_{s-w}$*

Table 3.11: Subcategorisation frame types: VPI

Frame Type	Example
n	*Mein Vater$_n$ bleibt Lehrer.*
i	*Ihn zu verteidigen$_i$ ist Dummheit.*
s-dass	*Dass ich ihn treffe$_{s-dass}$, ist mir peinlich.*
s-ob	*Ob sie kommt$_{s-ob}$, ist unklar.*
s-w	*Wann sie kommt$_{s-w}$, wird bald klarer.*

Table 3.12: Subcategorisation frame types: VPK

Generalised Verb Phrase	Verb Phrase Type with Frame Type
S.n	VPA.n
S.na	VPA.na, VPP.n, VPP.nP
S.nd	VPA.nd, VPP.d, VPP.dP
S.np	VPA.np, VPP.p, VPP.pP
S.nad	VPA.nad, VPP.nd, VPP.ndP
S.nap	VPA.nap, VPP.np, VPP.npP
S.ndp	VPA.ndp, VPP.dp, VPP.dpP
S.ni	VPA.ni, VPP.i, VPP.iP
S.nai	VPA.nai, VPP.ni, VPP.niP
S.ndi	VPA.ndi, VPP.di, VPP.diP
S.nr	VPA.nr
S.nar	VPA.nar
S.ndr	VPA.ndr
S.npr	VPA.npr
S.nir	VPA.nir
S.x	VPA.x
S.xa	VPA.xa
S.xd	VPA.xd
S.xp	VPA.xp
S.xr	VPA.xr
S.xs-dass	VPA.xs-dass
S.ns-2	VPA.ns-2, VPP.s-2, VPP.sP-2
S.nas-2	VPA.nas-2, VPP.ns-2, VPP.nsP-2
S.nds-2	VPA.nds-2, VPP.ds-2, VPP.dsP-2
S.nrs-2	VPA.nrs-2
S.ns-dass	VPA.ns-dass, VPP.s-dass, VPP.sP-dass
S.nas-dass	VPA.nas-dass, VPP.ns-dass, VPP.nsP-dass
S.nds-dass	VPA.nds-dass, VPP.ds-dass, VPP.dsP-dass
S.nrs-dass	VPA.nrs-dass
S.ns-ob	VPA.ns-ob, VPP.s-ob, VPP.sP-ob
S.nas-ob	VPA.nas-ob, VPP.ns-ob, VPP.nsP-ob
S.nds-ob	VPA.nds-ob, VPP.ds-ob, VPP.dsP-ob
S.nrs-ob	VPA.nrs-ob
S.ns-w	VPA.ns-w, VPP.s-w, VPP.sP-w
S.nas-w	VPA.nas-w, VPP.ns-w, VPP.nsP-w
S.nds-w	VPA.nds-w, VPP.ds-w, VPP.dsP-w
S.nrs-w	VPA.nrs-w
S.k	VPK.n, VPK.i, VPK.s-dass, VPK.s-ob, VPK.s-w

Table 3.13: Generalised frame description

Clause Type	Example
verb first clause	*Liebt* Peter seine Freundin? Hat Peter seine Freundin *geliebt*?
verb second clause	Peter *liebt* seine Freundin. Peter hat seine Freundin *geliebt*.
verb final clause	weil Peter seine Freundin *liebt* weil Peter seine Freundin *geliebt* hat
relative clause	der seine Freundin *liebt* der seine Freundin *geliebt* hat
indirect *wh*-question	wer seine Freundin *liebt* wer seine Freundin *geliebt* hat
non-finite clause	seine Freundin zu *lieben* seine Freundin *geliebt* zu haben

Table 3.14: Clause type examples

As mentioned before, the lexical verb head as the bearer of the clausal subcategorisation needs to be propagated through the parse tree, since the head information is crucial for the argument selection. The grammar structures are therefore based on a so-called 'collecting strategy' around the verb head: The collection of verb adjacents starts at the verb head and is performed differently according to the clause type, since the verb complex is realised by different formations and is situated in different positions in the topological sentence structure. Table 3.14 illustrates the proposition *Peter liebt seine Freundin* 'Peter loves his girl-friend' in the different clause types with and without auxiliary verb. For example, in a verb first clause with the verb head as the finite verb, the verb head is in sentence initial position and all arguments are to its right. But in a verb first clause with the auxiliary verb as the finite verb, the verb head is in sentence final position and all arguments are between the auxiliary and the verb head.

Below, A to E describe the collecting strategies in detail. Depending on the clause type, they start collecting arguments at the lexical verb head and propagate the lexical head up to the clause type level, as the head superscripts illustrate. The clause type `S` indicates the frame type of the respective sentence. Adverbial and prepositional phrase adjuncts might be attached at all levels, without having impact on the strategy or the frame type. The embedding of `S` under `TOP` is omitted in the examples.

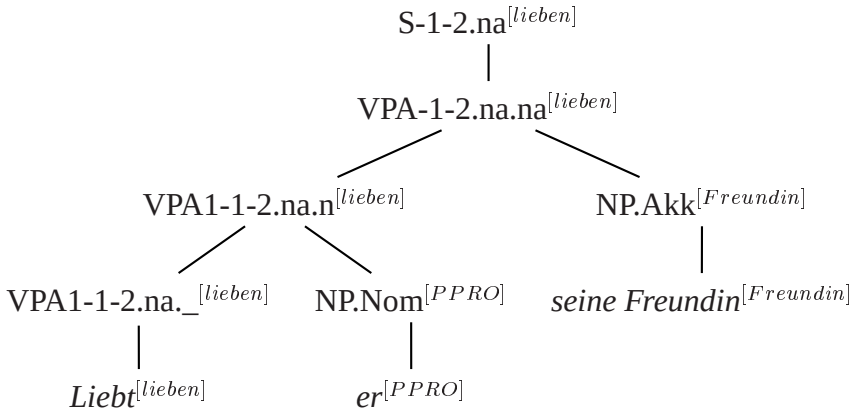
A Verb First and Verb Second Clauses

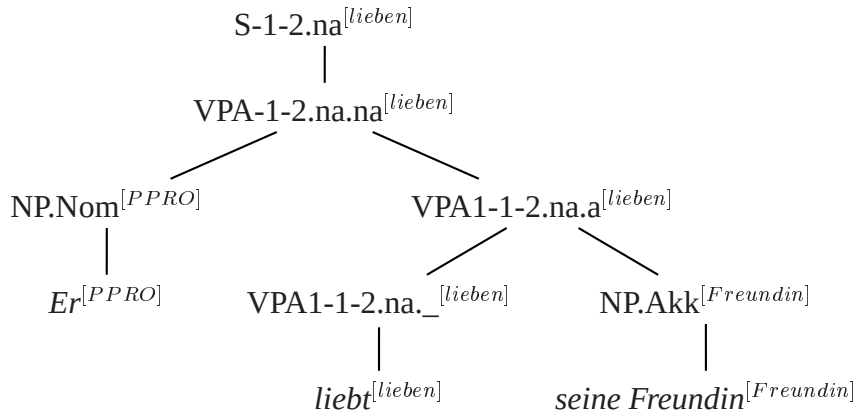
Verb first and verb second clauses are parsed by a common collecting schema, since they are similar in sentence formation and lexical head positions. The schema is sub-divided into three strategies:

(i) In clauses where the lexical verb head is expressed by a finite verb, the verb complex is identified as this finite verb and collects first all arguments to the right (corresponding to *Mittelfeld* and *Nachfeld* constituents) and then at most one argument to the left

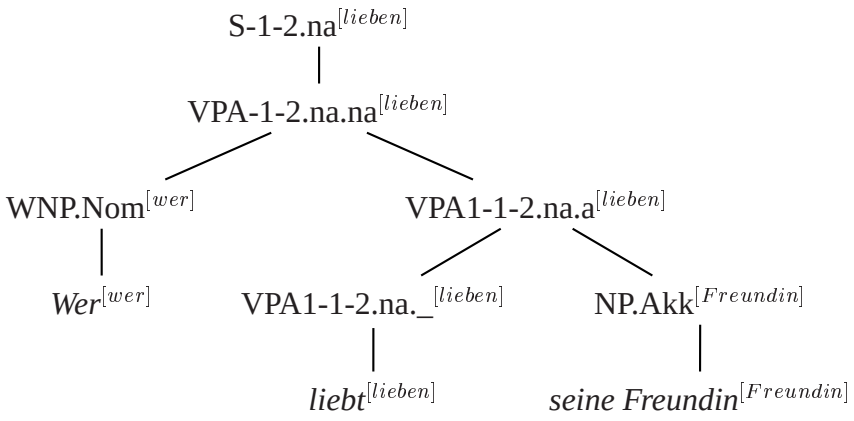
(corresponding to the *Vorfeld* position which is relevant for arguments in verb second clauses).

Below you find examples for both verb first and verb second clause types. The verb phrase annotation indicates the verb phrase type (VPA in the following examples), the clause type 1-2, the frame type (here: na) and the arguments which have been collected so far (_ for none). The 1 directly attached to the verb phrase type indicates the not yet completed frame. As verb first clause example, I analyse the sentence *Liebt er seine Freundin?* 'Does he love his girl-friend?', as verb second clause example, I analyse the sentence *Er liebt seine Freundin* 'He loves his girl-friend'. The lexical head of pronouns is PPRO.

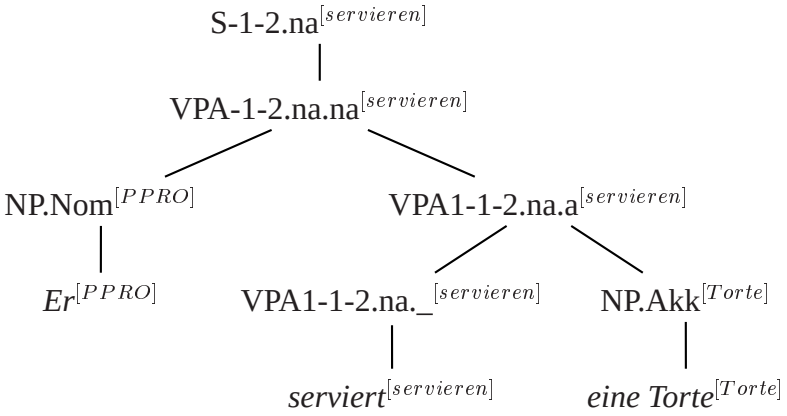




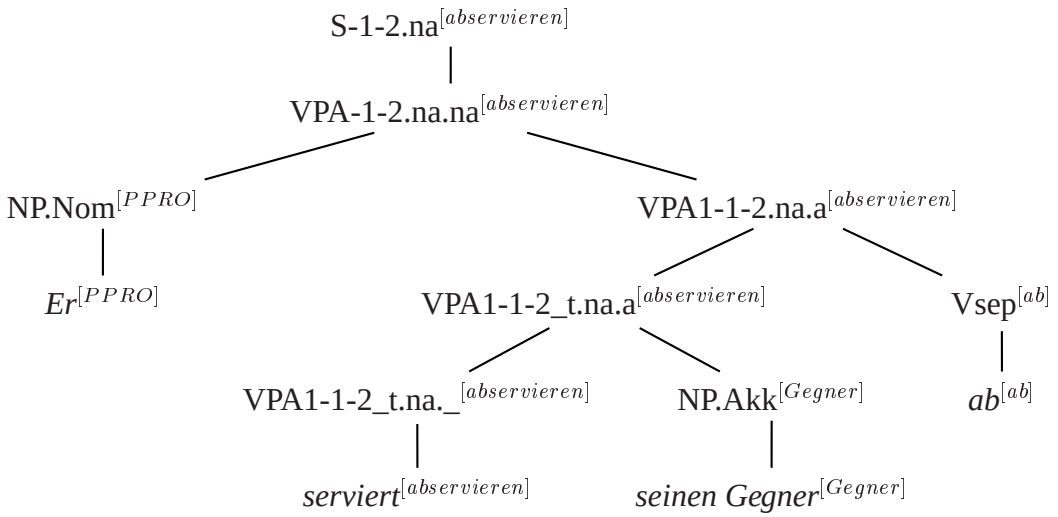
wh-questions are parsed in the same way as verb second clauses. They only differ in that the *Vorfeld* element is realised by a *wh*-phrase. The following parse tree analyses the question *Wer liebt seine Freundin?* 'Who loves his girl-friend?'. (Notice that *wh*-words in German are actually *w*-words.)



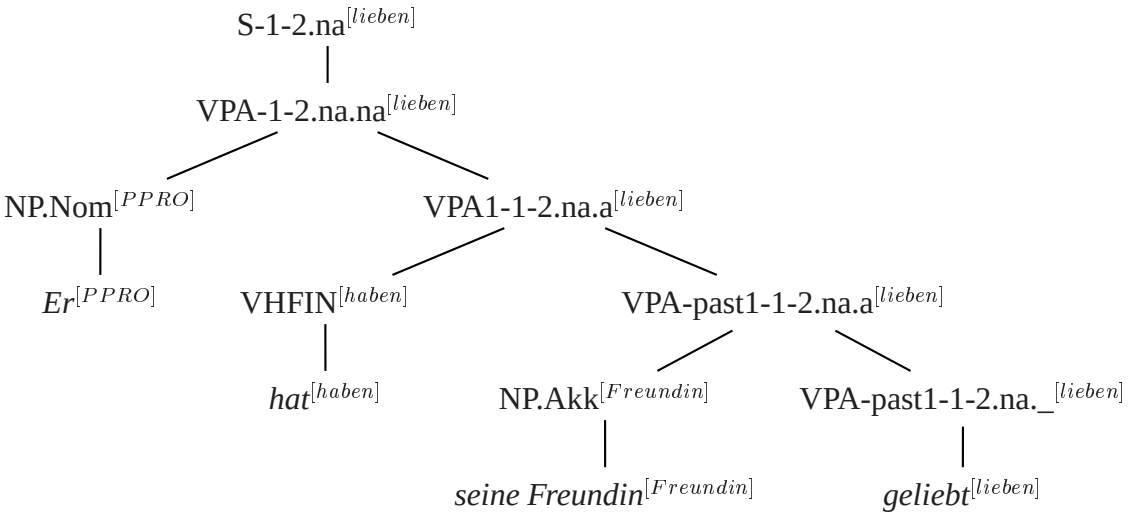
(ii) Finite verbs with separable prefixes collect their arguments in the same way. The notation differs in an additional indicator `_t` (for *trennbar* 'separable') which disappears as soon as the prefix is collected and the lexical head identified. It is necessary to distinguish verbs with separable prefixes, since the lexical verb head is only complete with the additional prefix. In this way we can, for example, differentiate the lexical verb heads *servieren* 'serve' and *abservieren* 'throw out' in *er serviert eine Torte* 'he serves a cake' vs. *er serviert seinen Gegner ab* 'he throws out his opponent'.[4] Following you find an example for the distinction. The head of the first tree is *servieren*, the head of the second tree *abservieren*:

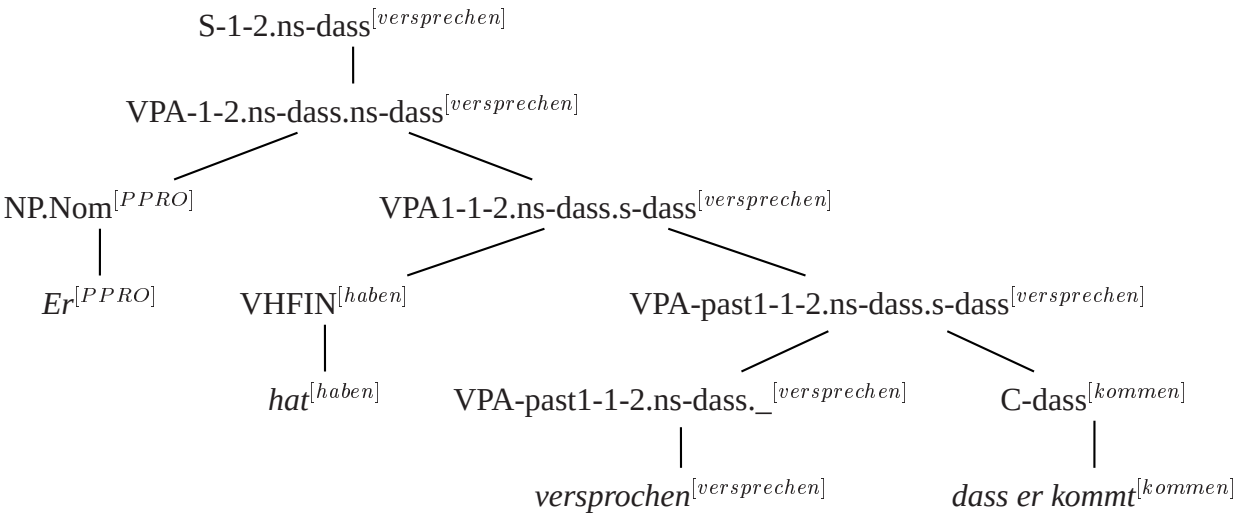


[4] `LoPar` provides a functionality to deal with particle verb lemmas.



(iii) In constructions with auxiliary verbs, the argument collection starts at the non-finite (participle, infinitival) lexical verb head, collecting arguments only to the left, since all arguments are defined in the *Vorfeld* and *Mittelfeld*. An exception to this rule are finite and non-finite clause arguments which can also appear in the *Nachfeld* to the right of the lexical verb head. The non-finite status of the verb category is marked by the low-level verb phrase types: `part` for participles and `inf` for infinitives. As soon as the finite auxiliary is found, at most one argument (to the left) is missing, and the non-finite marking on the clause category is deleted, to proceed as in (i). Below you find examples for verb second clauses: *Er hat seine Freundin geliebt* 'He has loved his girl-friend' and *Er hat versprochen, dass er kommt* 'He has promised to come'. The comma in the latter analysis is omitted for space reasons.

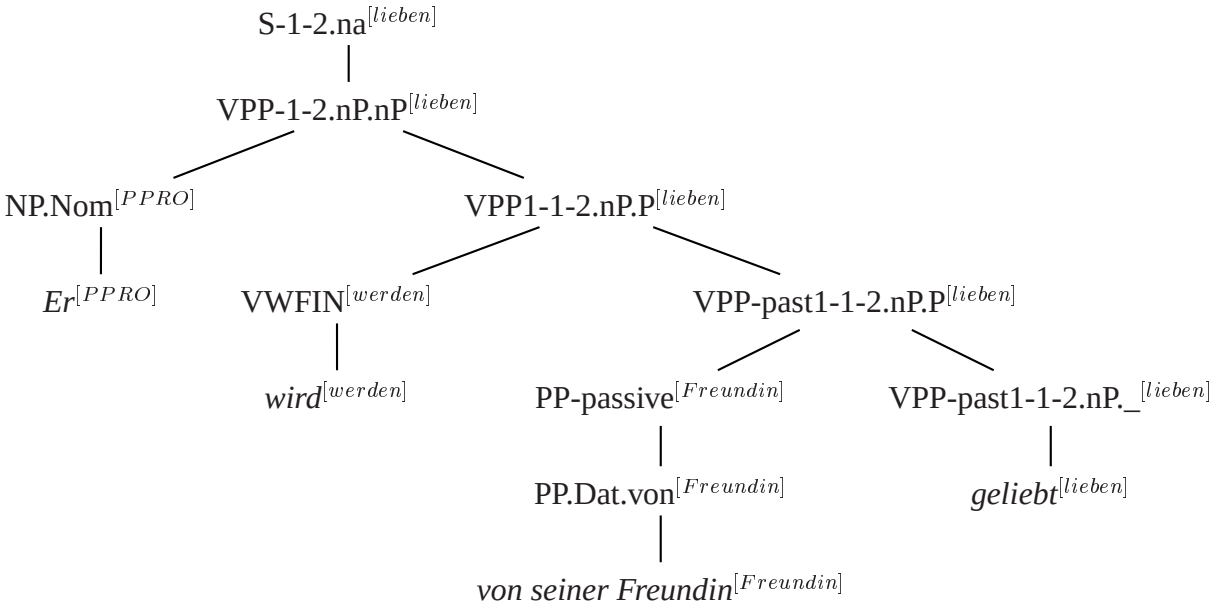




Strategies (i) and (ii) can only be applied to sentences without auxiliaries, which is a subset of VPA. Strategy (iii) can be applied to active and passive verb phrases as well as copula constructions. Table 3.15 defines the possible combinations of finite auxiliary verb and non-finite verb for the use of present perfect tense, passive voice, etc. An example analysis is performed for the sentence *Er wird von seiner Freundin geliebt* 'He is loved by his girl-friend'.

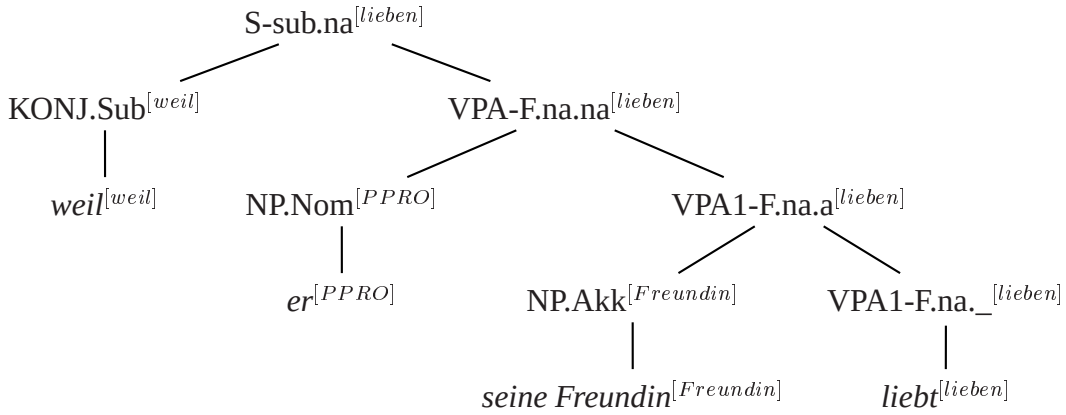
VP Type	Combination			Example
	Type	Auxiliary	Non-Finite Verb	
VPA	present perfect	VHFIN	past participle	*hat ... geliebt*
	present perfect	VSFIN	past participle	*ist ... geschwommen*
	'to have to, must'	VHFIN	infinitive	*hat ... zu bestehen*
	future tense	VWFIN	infinitive	*wird ... erkennen*
	modal construction	VMFIN	infinitive	*darf ... teilnehmen*
VPP	dynamic passive	VWFIN	past participle	*wird ... gedroht*
	statal passive	VSFIN	past participle	*ist ... gebacken*
	modal construction	VMFIN	past participle	*möchte ... geliebt werden*
VPK	'to be'	VSFIN	predicative	*ist ... im 7. Himmel*
	'to become'	VWFIN	predicative	*wird ... Lehrer*
	'to remain'	VBFIN	predicative	*bleibt ... doof*

Table 3.15: Auxiliary combination with non-finite verb forms

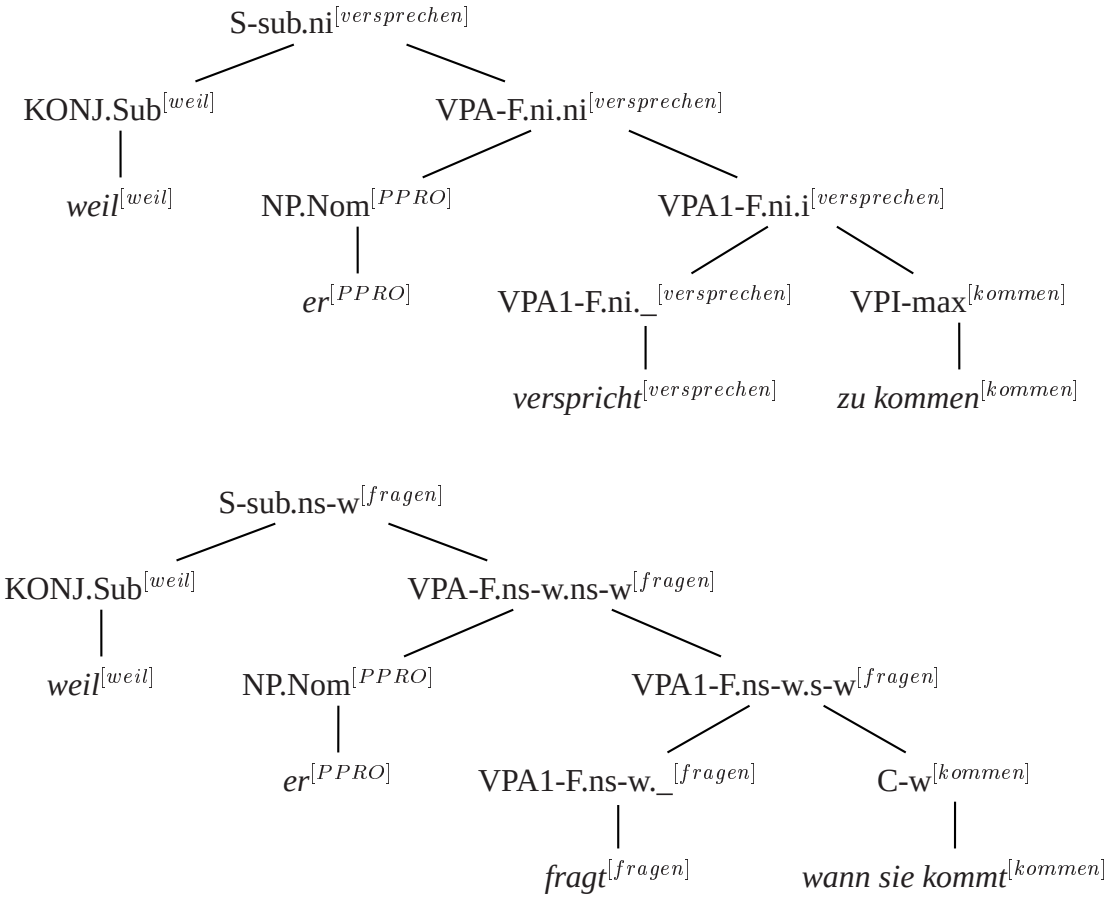


B Verb Final Clauses

In verb final clauses, the lexical verb complex is in the final position. Therefore, verb arguments are collected to the left only, starting from the finite verb complex. The verb final clause type is indicated by F. An example analysis for the sub-ordinated sentence *weil er seine Freundin liebt* 'because he loves his girl-friend' is given.

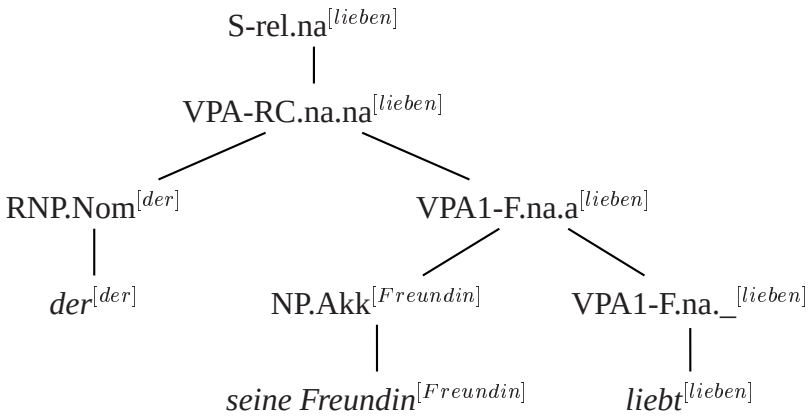


As an exception to the collecting strategy, clausal arguments might appear in the *Nachfeld* to the right of the verb complex. Below, two examples are given: In *weil er verspricht zu kommen* 'because he promises to come', *verspricht* in final clause position subcategorises a non-finite clause (`VPI-max` is a generalisation over all non-finite clauses), and in *weil er fragt, wann sie kommt* 'because he asks when she is going to come', *fragt* in clause final position subcategorises a finite *wh*-clause. The comma in the latter analysis is omitted for space reasons.



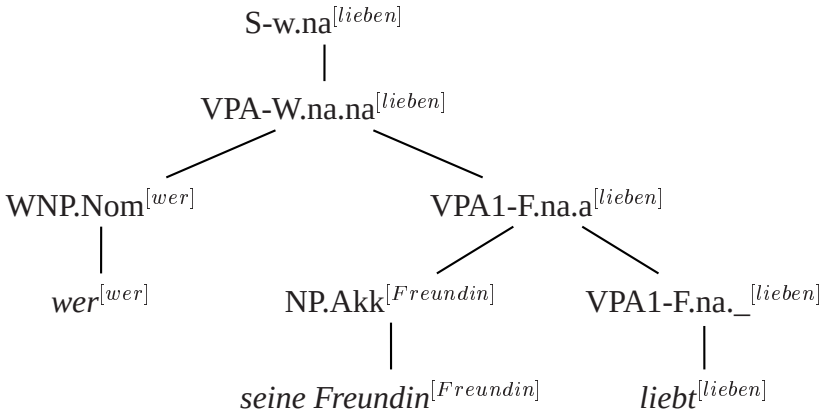
C Relative Clauses

Relative clauses are verb final clauses where the leftmost argument to collect is a noun phrase, prepositional phrase or non-finite clause containing a relative pronoun: `RNP`, `RPP`, `VPI-RC-max`. The clause type is indicated by `F` (as for verb final clauses) until the relative pronoun phrase is collected; then, the clause type is indicated by `RC`. An example analysis is given for *der seine Freundin liebt* 'who loves his girl-friend'. As for verb final clauses, finite and non-finite clauses might be subcategorised to the right of the finite verb.



D Indirect *wh*-Questions

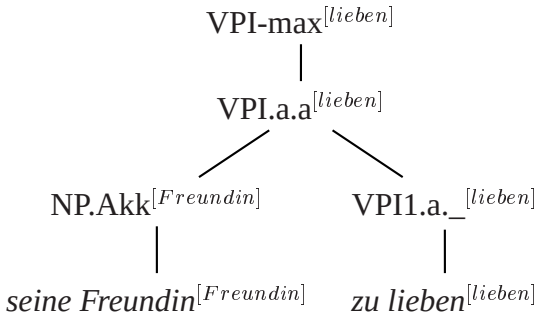
Indirect *wh*-questions are verb final clauses where the leftmost argument to collect is a noun phrase, a prepositional phrase, an adverb, or a non-finite clause containing a *wh*-phrase: `WNP`, `WPP`, `WADV`, `VPI-W-max`. The clause type is indicated by `F` (as for verb final clauses) until the *wh*-phrase is collected; then, the clause type is indicated by `W`. An example analysis is given for *wer seine Freundin liebt* 'who loves his girl-friend'. As for verb final clauses, finite and non-finite clauses might be subcategorised to the right of the finite verb.



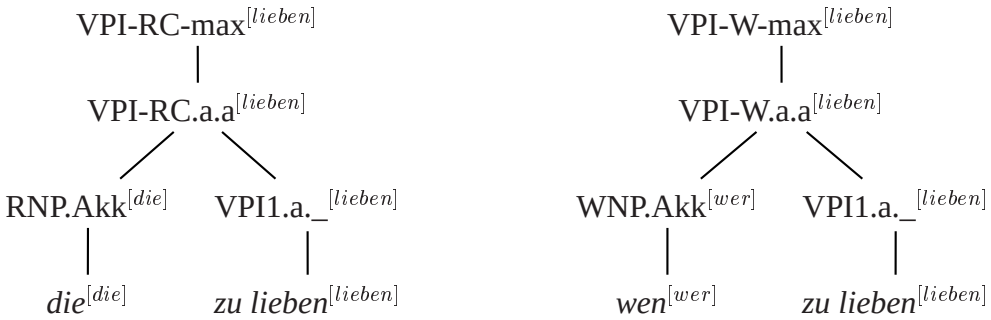
E Non-Finite Clauses

Non-finite clauses start collecting arguments from the non-finite verb complex and collect to the left only. As an exception, again, clausal arguments are collected to the right. An example analysis is given for *seine Freundin zu lieben* 'to love his girl-friend'. As mentioned before,

`VPI-max` is a generalisation over all non-finite clauses. It is the relevant category for the subcategorisation of a non-finite clause.



Non-finite clauses might be the introductory part of a relative clause or a *wh*-question. In that case, the leftmost argument contains a relative pronoun or a *wh*-phrase, and the `VPI` category is marked by `RC` or `W`, respectively. The following examples analyse *die zu lieben* 'whom to love' and *wen zu lieben* 'whom to love'.



Noun Phrases The noun phrase structure is determined by practical needs: Noun phrases are to be recognised reliably, and nominal head information has to be passed through the nominal structure, but the noun phrase structure is kept simple without a theoretical claim. There are four nominal levels: the terminal noun `NN` is possibly modified by a cardinal number `CARD`, a genitive noun phrase `NP.Gen`, a prepositional phrase adjunct `PP-adjunct`, a proper name phrase `NEP`, or a clause `S-NN`, and is dominated by `N1`. `N1` itself may be modified by an (attributive) adjectival phrase `ADJaP` to reach `N2` which can be preceded by a determiner (`ART, DEM, INDEF, POSS`) to reach the `NP` level. All noun phrase levels are accompanied by the case feature. Figure 3.8 describes the noun phrase structure, assuming case agreement in the constituents. The clause label `S-NN` is a generalisation over all types of clauses allowed as noun modifier: `C-rel, C-dass, C-ob, C-w`. Example analyses are provided for the noun phrases *jener Mann mit dem Hut* 'that man with the hat'$_{Nom}$, *den alten Bauern Fehren* 'the old farmer Fehren'$_{Akk}$, and *der Tatsache, dass er schläft* 'the fact that he sleeps'$_{Gen}$.

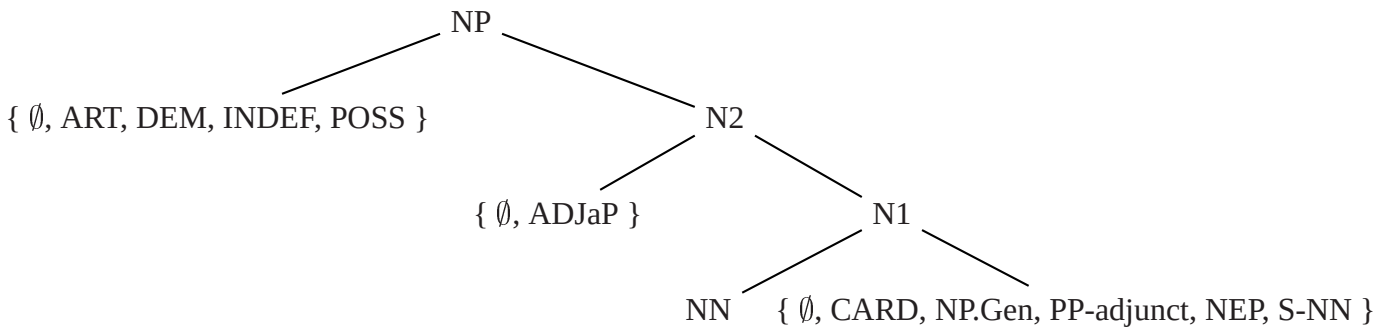
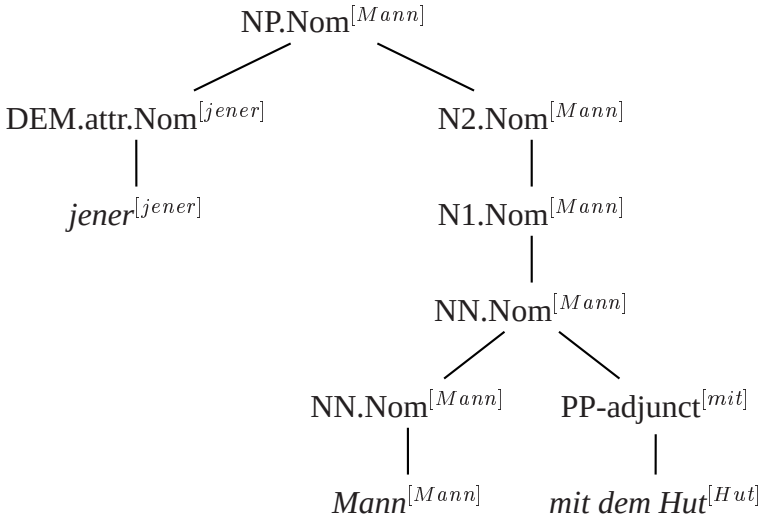
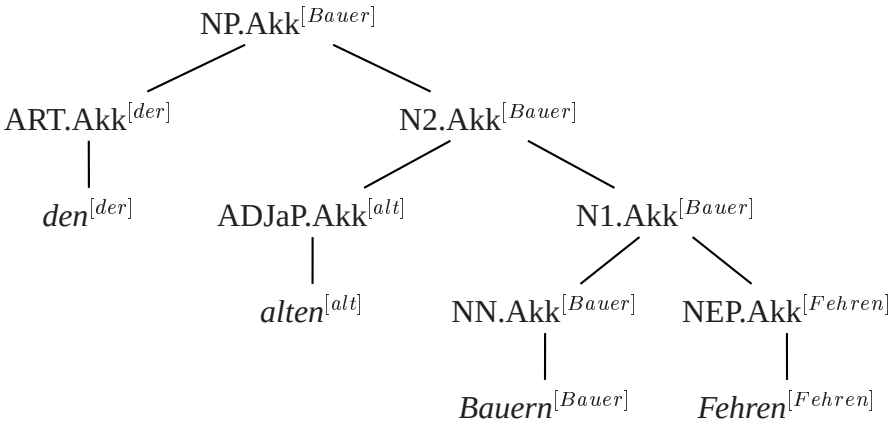


Figure 3.8: Nominal syntactic grammar categories





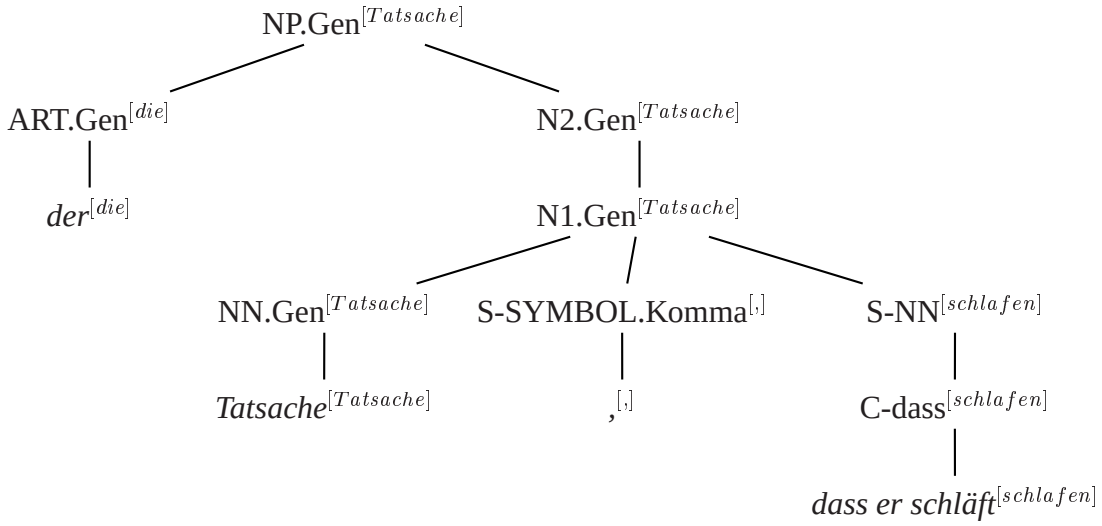


Figure 3.9 describes that proper name phrases NEP are simply defined as a list of proper names. As for common nouns, all levels are equipped with the case feature. Example analyses are provided for *New York*$_{Akk}$ and *der alte Peter* 'the old Peter'$_{Nom}$.

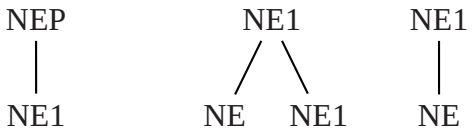


Figure 3.9: Proper names

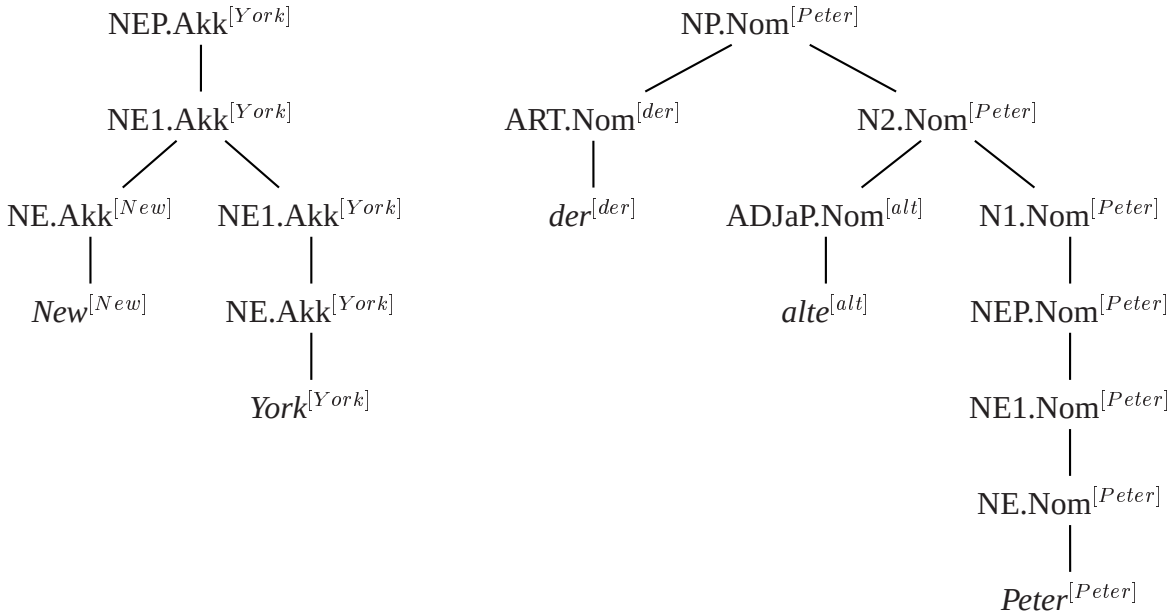


Figure 3.10 shows that noun phrases can generate pronouns and cardinal numbers, which do not allow modifiers. A number of examples is provided, illustrating the simple analyses for *ich* 'I'$_{Nom}$, *dich* 'you'$_{Akk}$, *einander* 'each other'$_{Akk}$, and *allen* 'everybody'$_{Dat}$.

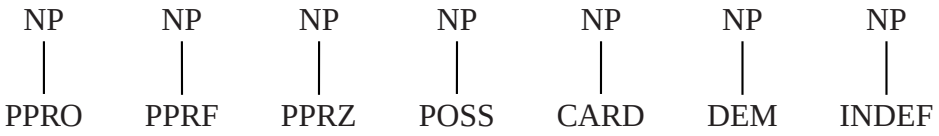
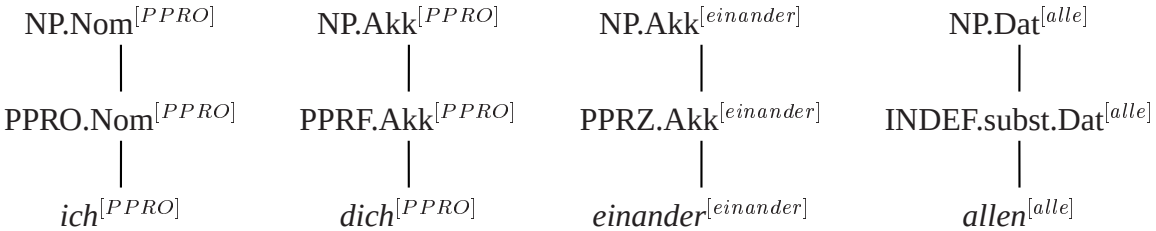


Figure 3.10: Noun phrases generating pronouns and cardinals



For relative and interrogative clauses, the specific kinds of NPs introducing the clause need to be defined, either as stand-alone pronoun, or attributively combined with a nominal on N2 level. RNP and WNP are also equipped with the case feature. See the definition in Figure 3.11 and a number of example analyses for *der* 'who'$_{Nom}$, *dessen Bruder* 'whose brother'$_{Akk}$, and *wem* 'whom'$_{Dat}$.

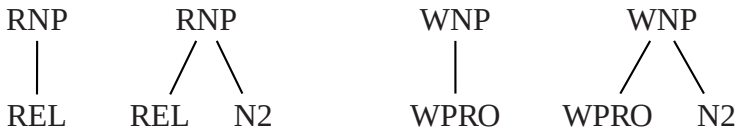
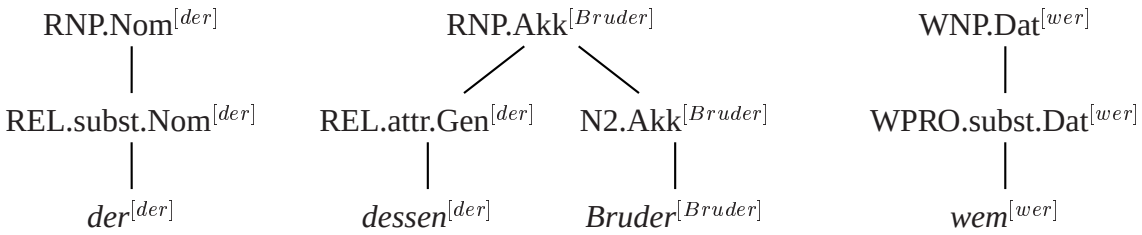


Figure 3.11: Noun phrases introducing relative and interrogative clauses



Prepositional Phrases Prepositional phrases are distinguished in their formation with respect to their syntactic function: (A) arguments vs. (B) adjuncts. By introducing both PP-arguments and PP-adjuncts I implicitly assume that the statistical grammar model is able to learn the distinction between the grammatical functions. But this distinction raises two questions:

1. Which is the distinction between PP-arguments and PP-adjuncts?

   As mentioned before, to distinguish between arguments and adjuncts I refer to the optionality of the complements. But with prepositional phrases, there is more to take into consideration. Standard German grammar such as Helbig and Buscha (1998, pages 402–404) categorise adpositions with respect to their usage in argument and adjunct PPs. With respect to PP-arguments, we distinguish verbs which are restricted to a single adposition as head of the PP (such as *achten auf* 'to pay attention, to look after') and verbs which require a PP of a certain semantic type, but the adpositions might vary (e.g. *sitzen* 'to sit' requires a local PP which might be realised by prepositions such as *auf, in*, etc.). Adpositions in the former kind of PP-arguments lose their lexical meaning in composition with a verb, so the verb-adposition combination acquires a non-compositional, idiosyncratic meaning. Typically, the complements of adpositions in PP-arguments are more restricted than in PP-adjuncts.

2. Is it possible to learn the distinction between PP-arguments and PP-adjuncts?

   To learn the distinction between PP-arguments and PP-adjunct is a specifically hard problem, because structurally each PP in the grammar can be parsed as argument and as adjunct, as the PP-implementation below will illustrate. The clues for the learning therefore lie in the distinction of the lexical relationships between verbs and adpositions and verbs and PP-subcategorised (nominal) head. The lexical distinction is built into the grammar rules as described below and even though not perfect actually helps the learning (cf. the grammar evaluation in Section 3.5).

A PP-Arguments

Prepositional phrase arguments combine the generated adposition with case information, i.e. `PP.<case>.<adposition>`. Basically, their syntactic structure requires an adposition or a comparing conjunction, and a noun or adverbial phrase, as Figure 3.12 shows. The head of the PP-argument is defined as the head of the nominal or adverbial phrase subcategorised by the adposition. By that, the definition of PP-arguments provides both the head information of the adposition in the category name (to learn the lexical relationship between verb and adposition) and the head information of the subcategorised phrase (to learn the lexical relationship between verb and PP-subcategorised nominal or adverbial head). Examples for the prepositional phrases in Figure 3.12 are *wie ein Idiot* 'as an idiot', *von drüben* 'from over there', *am Hafen* 'at the port', *meiner Mutter wegen* 'because of my mother'.

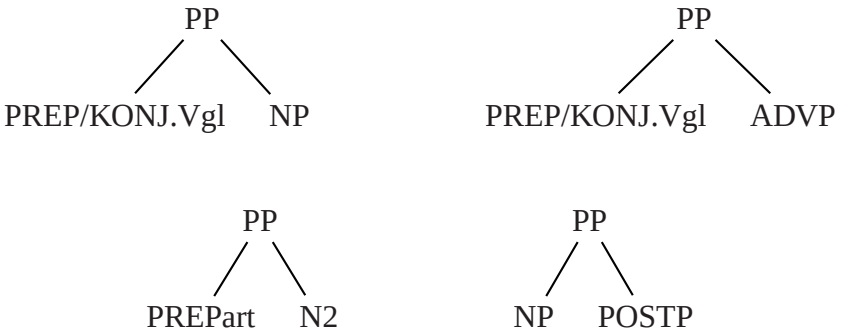
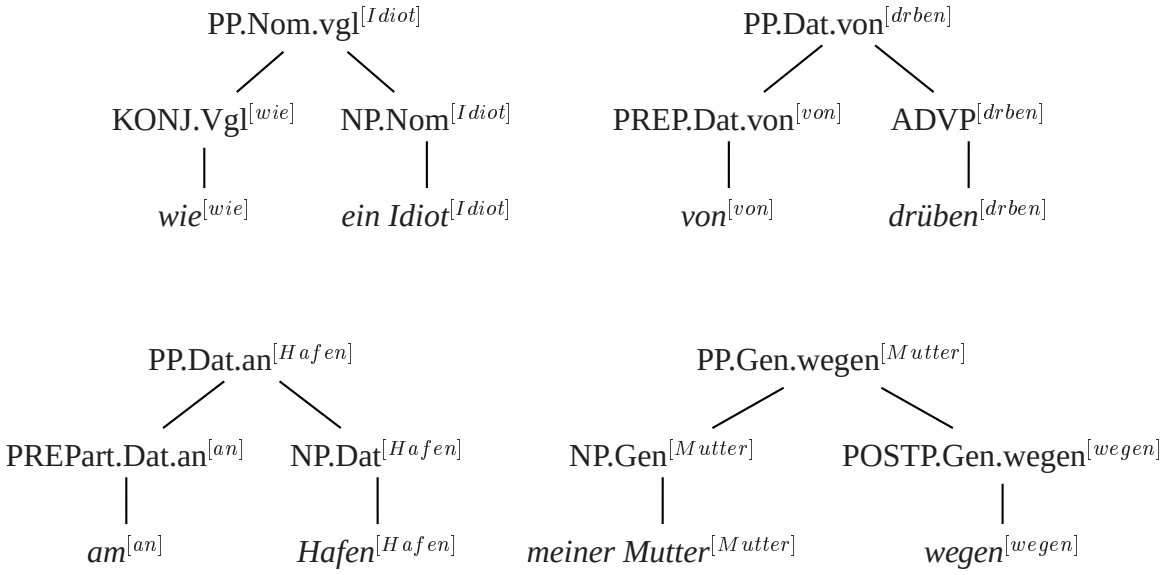


Figure 3.12: Prepositional phrase arguments



In addition, the prepositional phrases generate pronominal and interrogative adverbs if the preposition is the morphological head of the adverb, for example:

```
PP.Akk.für -> PROADV.dafür'
```

Like for noun phrases, the specific kinds of PP-arguments which introduce relative and interrogative clauses need to be defined. See the definition in Figure 3.13. Examples are given for *mit dem* 'with whom', *durch wessen Vater* 'by whose father', and *wofür* 'for what'.

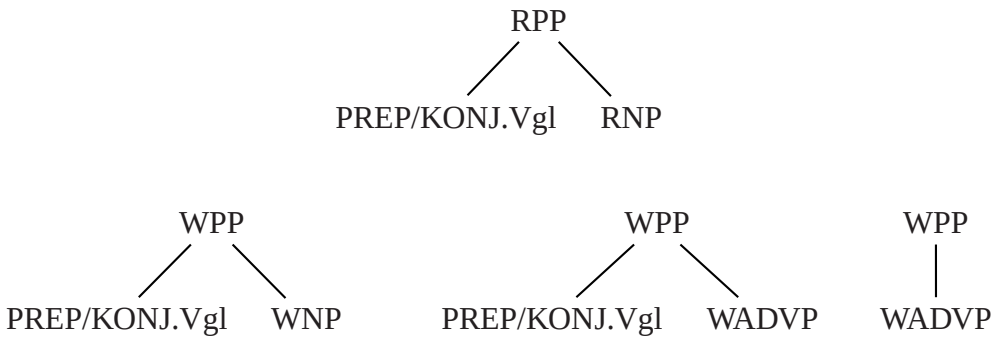
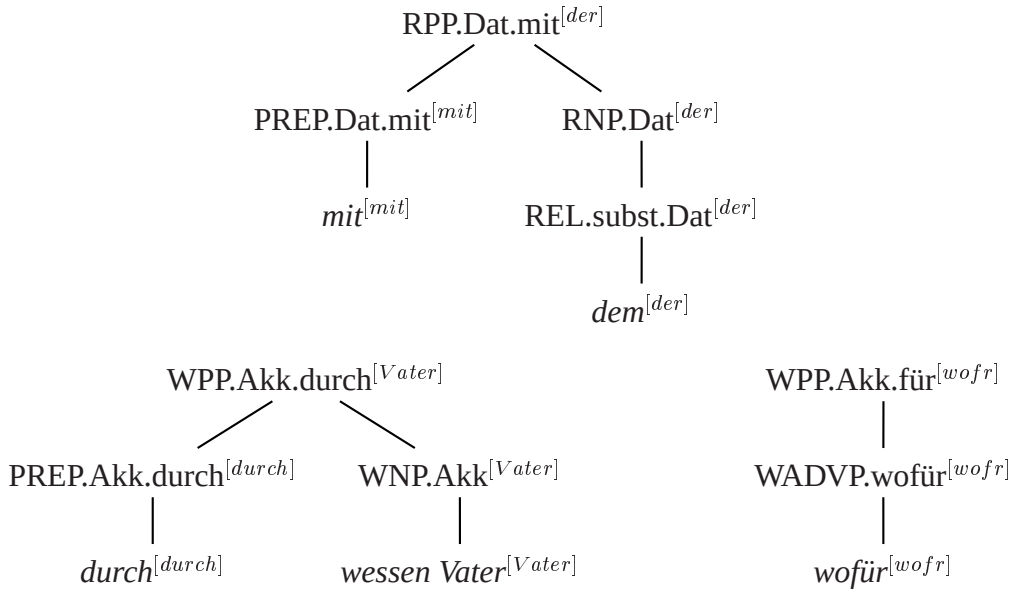
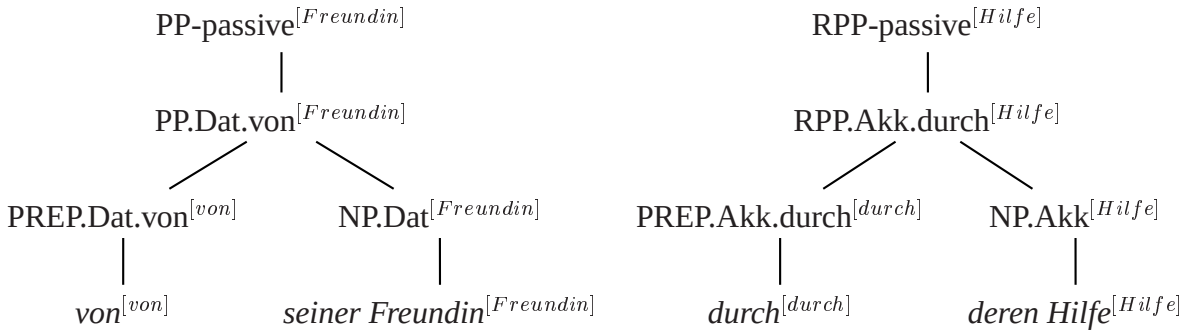


Figure 3.13: Prepositional phrase arguments in relative and interrogative clauses



Finally, a syntactically based category `(R/W)PP-passive` generates the two prepositional phrases `(R/W)PP.Akk.durch` and `(R/W)PP.Dat.von` as realisations of the deep structure subject in passive usage. See the examples for *von seiner Freundin* 'by his girl-friend', and *durch deren Hilfe* 'by the help of who'.



B PP-Adjuncts

Prepositional phrase adjuncts are identified by the syntactic category `(R/W)PP-adjunct`. As `PP`-arguments, they require an adposition and a noun or adverbial phrase (cf. Figure 3.14), but the head of the PP-adjunct is the adposition, because the information subcategorised by the adposition is not considered relevant for the verb subcategorisation. Example analyses are provided for *bei dem Tor* 'at the gate', *nach draußen* 'to the outside', and *zu dem* 'towards who'.

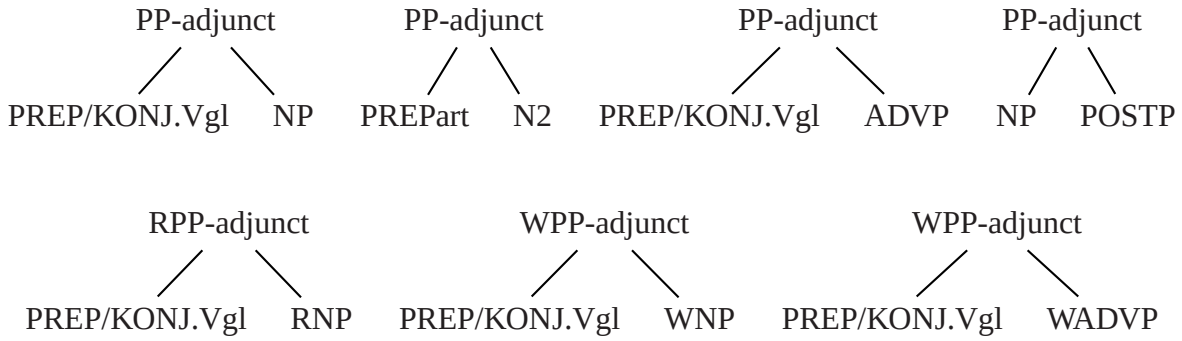
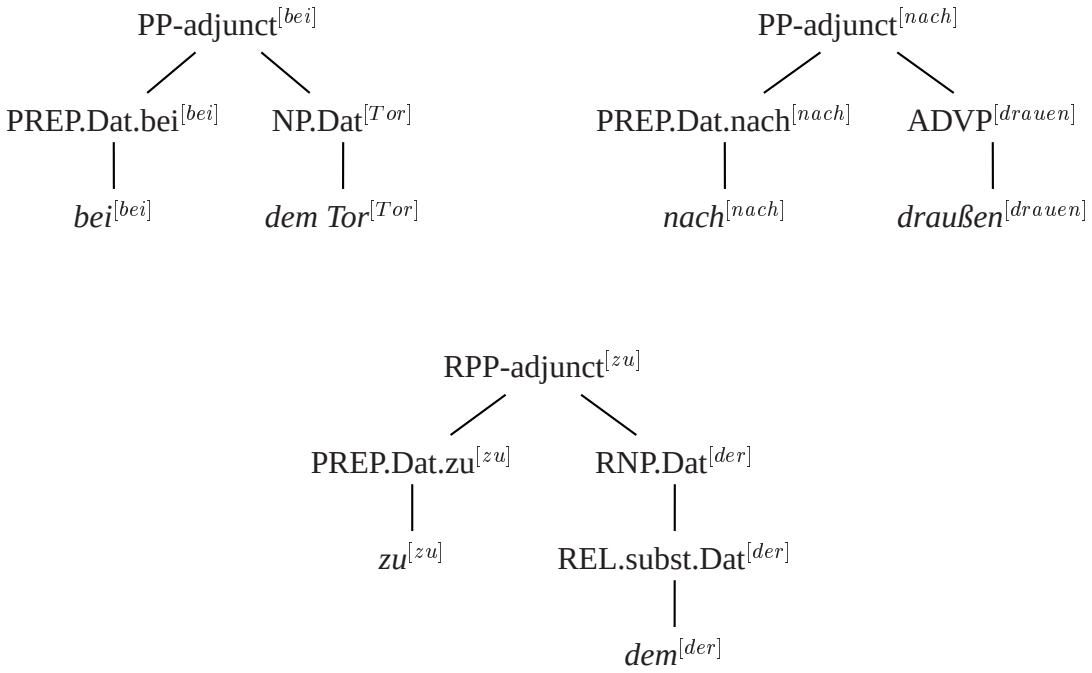


Figure 3.14: Prepositional phrase adjuncts



Adjectival Phrases Adjectival phrases distinguish between (A) an attributive and (B) a predicative usage of the adjectives.

A Attributive Adjectives

Attributive adjectival phrases are realised by a list of attributive adjectives. The adjectives are required to agree in case. Terminal categories other than declinable attributive adjectives are indeclinable adjectives, and cardinal and ordinal numbers. The attributive adjective formation is illustrated in Figure 3.15. Attributive adjectives on the bar level might be combined with adverbial adjuncts. Example analyses are provided for *tollen alten* 'great old'$_{Akk}$, and *ganz lila* 'completely pink'$_{Nom}$.

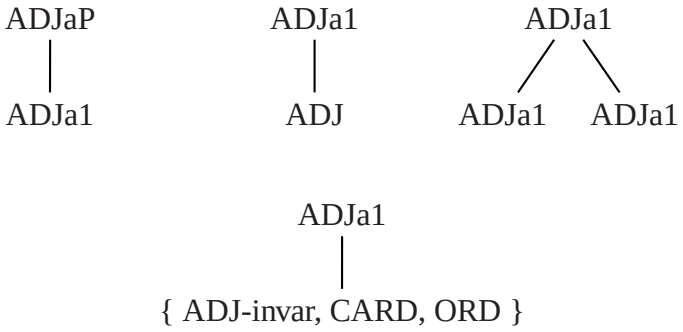
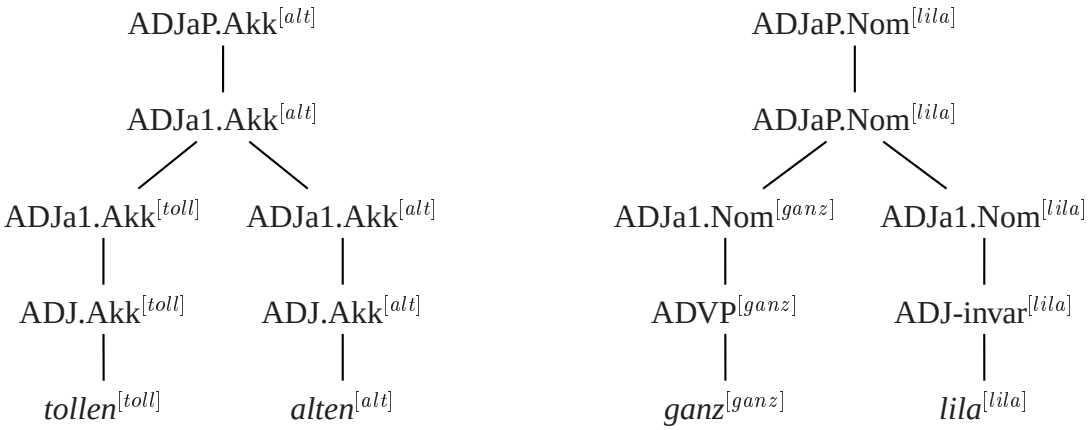


Figure 3.15: Attributive adjectival phrases



B Predicative Adjectives

Predicative adjectival phrases are realised by a predicative adjective (possibly modified by a particle), or by an indeclinable adjective, as displayed by Figure 3.16. As attributive adjectival phrases, the predicative adjectives on the bar level might be combined with adverbial adjuncts. Example analyses are given for *zu alt* 'too old' and *wirklich hervorragend* 'really excellent'.

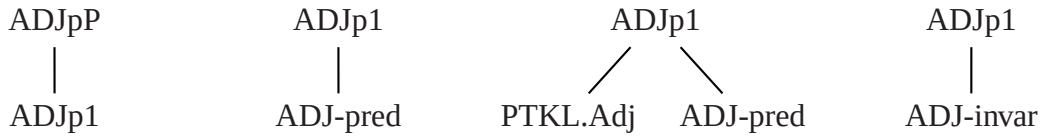
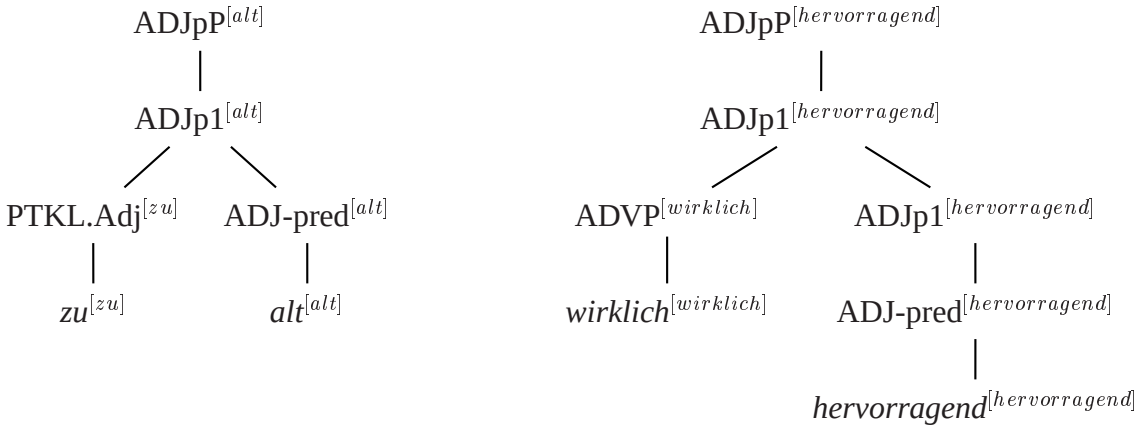
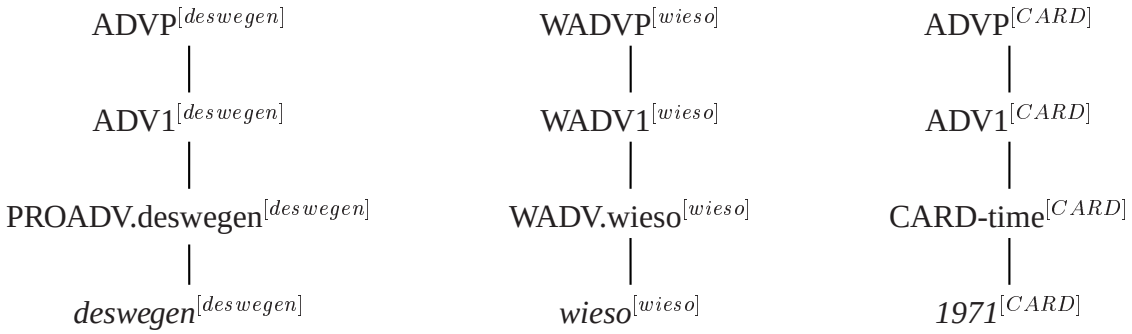


Figure 3.16: Predicative adjectival phrases



Adverbial Phrases Adverbial phrases (W)ADVP are realised by adverbs, pronominal or interrogative adverbs. Terminal categories other than adverbs are predicative adjectives, particles, interjections, and year numbers. The adverbial formation is illustrated in Figure 3.17, and examples are provided for *deswegen* 'because of that', *wieso* 'why', and *1971*. The lexical head of cardinal numbers is CARD.



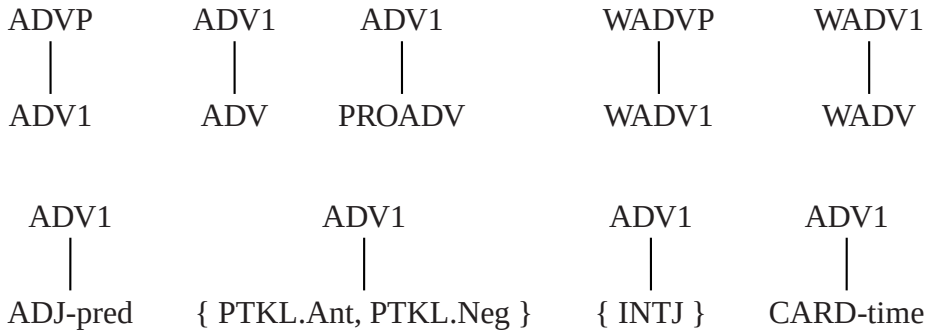
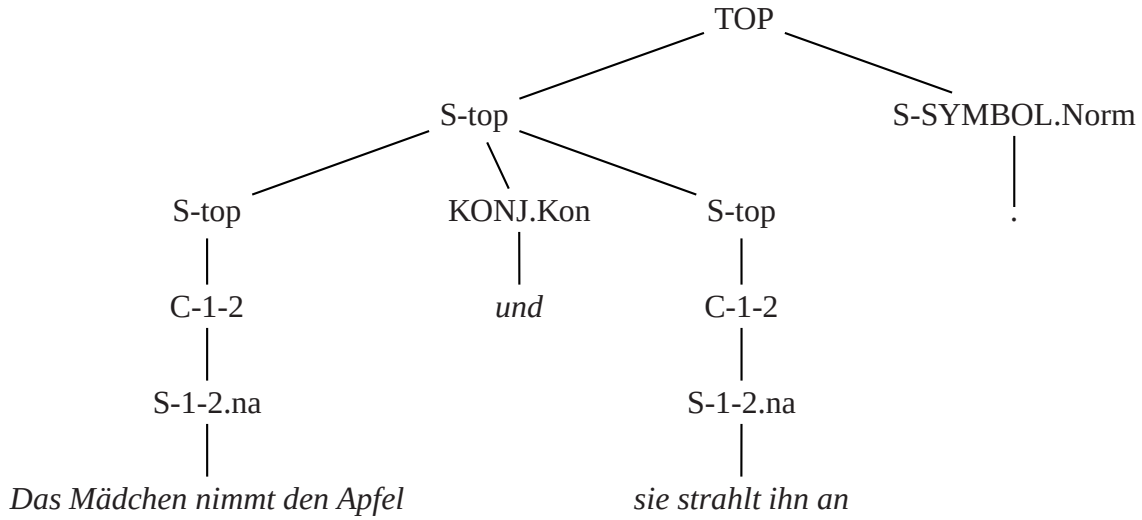


Figure 3.17: Adverbial phrases

Coordination Since coordination rules extensively inflate the grammar, coordination is only applied to specific grammar levels. Noun phrases, prepositional phrases, adjectival and adverbial phrases are combined on the phrase level only. For example, the structurally ambiguous NP *die alten Männer und Frauen* 'the old men and women' is analysed as [[die alten Männer]$_{NP}$ & [Frauen]$_{NP}$], but not as [die [alten Männer]$_{N2}$ & [Frauen]$_{N2}$] or [die alten [Männer$_{N1}$ & Frauen$_{N1}$]], since coordination only applies to NP, but not to N2 or N1. Coordination of verb units is performed on fully saturated verb phrases (via the `S-top` level) and on verb complexes. For example, the grammar fails in parsing *Das Mädchen nimmt den Apfel und strahlt ihn an* 'the girl takes the apple and smiles at him', because it would need to combine a fully saturated `VPA.na` with a `VPA.na` missing the subject. In contrast, the grammar is able to parse *Das Mädchen nimmt den Apfel und sie strahlt ihn an* 'the girl takes the apple and she smiles at him', because it combines two fully saturated `VPA.na` at the `S-top` level:



The restriction on coordination is a compromise between the necessity of including coordination into the grammar and the large number of parameters resulting from integrating coordination for all possible categories and levels, especially with respect to the fine-grained subcategorisation information in the grammar.

3.3 Grammar Training

The previous section has described the development and implementation of the German context-free grammar. This section uses the context-free backbone as basis for the lexicalised probabilistic extension, to learn the statistical grammar model. The grammar training is performed by the statistical parser LoPar (Schmid, 2000). Section 3.3.1 introduces the key features of the parser, and Section 3.3.2 describes the training strategy to learn the statistical grammar model.

3.3.1 The Statistical Parser

LoPar is an implementation of the left-corner parsing algorithm. Its functionality comprises symbolic parsing with context-free grammars, and probabilistic training and parsing with probabilistic context-free grammars and head-lexicalised probabilistic context-free grammars. In addition, the parser can be applied for Viterbi parsing, tagging and chunking.

LoPar executes the parameter training of the probabilistic context-free grammars by the *Inside-Outside Algorithm* (Lari and Young, 1990), an instance of the *Expectation-Maximisation (EM) Algorithm* (Baum, 1972). The EM-algorithm is an unsupervised iterative technique for maximum likelihood approximation of training data. Each iteration in the training process consists of an estimation (E) and a maximisation (M) step. The E-step evaluates a probability distribution for the data given the model parameters from the previous iteration. The M-step then finds the new parameter set that maximises the probability distribution. So the model parameters are improved by alternately assessing frequencies and estimating probabilities. The EM-algorithm is guaranteed to find a local optimum in the search space. EM is sensitive to the initialisation of the model parameters. For the *Inside-Outside Algorithm*, the EM-parameters refer to grammar-specific training data, i.e. how to determine the probabilities of sentences with respect to a grammar. The training is based on the notion of grammar categories and estimates the parameters producing a category ('outside' the category with respect to a tree structure) and the parameters produced by a category ('inside' the category with respect to a tree structure), hence the name. The parameter training with LoPar is performed by first optimising the PCFG parameters, then using the PCFG parameters for a bootstrapping of the lexicalised H-L PCFG model, and finally optimising the H-L PCFG parameters.

According to Manning and Schütze (1999), a main problem of H-L PCFGs is that for discriminating the large number of parameters a sufficient amount of linguistic data is required. The sparse data problem is pervasive, so effective smoothing techniques are necessary. LoPar implements four ways of incorporating sparse data into the probabilistic model:

(i) The number of parameters is reduced by allowing lemmatised word forms instead of fully inflected word forms.

(ii) All unknown words are tagged with the single token <unknown> which also propagates as lexical head. A set of categories for unknown words may be determined manually before

the parsing process, e.g. noun tags are assigned by default to capitalised unknown words, and verb tags or adjective tags to non-capitalised unknown words. This handling prevents the parser from failing on sentences with unknown words.

(iii) Parameter smoothing is performed by absolute discounting. The smoothing technique as defined by Ney *et al.* (1994) subtracts a fixed discount from each non-zero parameter value and redistributes the mass of the discounts over unseen events.

(iv) The parameters of the head-lexicalised probabilistic context-free grammar can be manually generalised for reduction (see 'parameter reduction' below on details).

3.3.2 Training Strategy

The training strategy is the result of experimental work on H-L PCFGs for German, since there is no 'rule of thumb' for the parameter training which is valid for all possible setups. Former versions of the training setup and process are reported by Beil *et al.* (1999), Schulte im Walde (2000b) and Schulte im Walde *et al.* (2001). The latter reference contains an evaluation of diverse training strategies.

Training Corpus As training corpus for the German grammar model, I use parts of a large German newspaper corpus from the 1990s, which is referred to as the *Huge German Corpus (HGC)*. The HGC contains approximately 200 million words of newspaper text from *Frankfurter Rundschau*, *Stuttgarter Zeitung*, *VDI-Nachrichten*, *die tageszeitung*, *German Law Corpus*, *Donaukurier*, and *Computerzeitung*.

The corpus training data should be as numerous as possible, so the training should be performed on all 200 million words accessible. On the other hand, time constraints make it necessary to restrict the amount of data. The following training parameters have been developed out of experience and as a compromise between data and time demands.

- All 6,591,340 sentences (82,149,739 word tokens) from the HGC with a length between 5 and 20 words are used for unlexicalised training. The grammar has a coverage[5] of parsing 68.03% of the sentences, so effectively the training is performed on 4,484,089 sentences.
- All 2,426,925 sentences (18,667,888 word tokens) from the HGC with a length between 5 and 10 words are used for the lexicalisation, the bootstrapping of the lexicalised grammar model. The grammar has a coverage of parsing 71.75% of the sentences, so effectively the bootstrapping is performed on 1,741,319 sentences.
- All 3,793,768 sentences (35,061,874 word tokens) from the HGC with a length between 5 and 13 words are used for lexicalised training. The grammar has a coverage of parsing 71.74% of the sentences, so effectively the training is performed on 2,721,649 sentences.

[5]The *coverage* of the grammar refers to the percentage of sentences from the corpus which are assigned at least one parse analysis. The sentences without an analysis are not taken into consideration for in training process.

Initialisation and Training Iterations The initialisation of the PCFG grammar parameters is performed by assigning the same frequency to all grammar rules. Comparable initialisations with random frequencies had no effect on the model development (Schulte im Walde, 2000b). The parameter estimation is performed within one iteration for unlexicalised training of the PCFG, and three iterations for lexicalised training of the H-L PCFG. The overall training process takes 15 days on a Sun Enterprise 450 with 296 MHz CPU.

Parameter Reduction As mentioned before, `LoPar` allows a manual generalisation to reduce the number of parameters. The key idea is that lexical heads which are supposed to overlap for different grammar categories are tied together. For example, the direct objects of *kaufen* 'to buy' are the same irrespective of the degree of saturation of a verb phrase and also irrespective of the clause type. Therefore, I can generalise over the transitive verb phrase types `VPA1-1-2.na._, VPA1-1-2.na.n, VPA1.1-2.na.a, VPA-1-2.na.na` and include the generalisation over the different clause types `1-2, rel, sub, dass, ob, w`. In addition, we can generalise over certain arguments in active and passive and in finite and non-finite verb phrases, for example the accusative object in an active finite clause `VPA` for frame type `na` and the accusative object in an active non-finite clause `VPI` for frame type `a`. The generalisation is relevant for the lexicalised grammar model and is performed for all verb phrase types. The parameter reduction in the grammar is especially important because of the large number of subcategorisation rules.

Summary We can summarise the process of grammar development and training strategy in the following steps.

1. Manual definition of CFG rules with head-specification,
2. Assigning uniform frequencies to CFG rules (extension of CFG to PCFG),
3. Unlexicalised training of the PCFG: one iteration on approx. 82 million words,
4. Manual definition of grammar categories for parameter reduction,
5. Lexicalisation of the PCFG (bootstrapping of H-L PCFG) on approx. 19 million words,
6. Lexicalised training of the H-L PCFG: three iterations on approx. 35 million words.

3.4 Grammar-Based Empirical Lexical Acquisition

The previous sections in this chapter have introduced the German grammar implementation and training. The resulting statistical grammar model provides empirical lexical information, specialising on but not restricted to the subcategorisation behaviour of verbs. In the following, I present examples of such lexical information. The examples are selected with regard to the lexical verb descriptions at the syntax-semantic interface which I will use in the clustering experiments.

Section 3.4.1 describes the induction of subcategorisation frames for the verbs in the German grammar model, and Section 3.4.2 illustrates the acquisition of selectional preferences. In Section 3.4.3 I present related work on the automatic acquisition of lexical information within the framework of H-L PCFGs.

3.4.1 Subcategorisation Frames

The acquisition of subcategorisation frames is directly related to the grammar implementation. Recall the definition of clause types: The clause level C produces the clause category S which is accompanied by the relevant subcategorisation frame dominating the clause. Each time a clause is analysed by the statistical parser, a clause level rule with the relevant frame type is included in the analysis.

C-<type> $\rightarrow$ S-<type>.<frame>

The PCFG extension of the German grammar assigns frequencies to the grammar rules according to corpus appearance and is able to distinguish the relevance of different frame types. The usage of subcategorisation frames in the corpus is empirically trained.

freq_1 C-<type> $\rightarrow$ S-<type>.$\text{<frame}_1\text{>}$
freq_2 C-<type> $\rightarrow$ S-<type>.$\text{<frame}_2\text{>}$
$\text{freq}_{...}$ C-<type> $\rightarrow$ S-<type>.$\text{<frame}_{...}\text{>}$
freq_n C-<type> $\rightarrow$ S-<type>.$\text{<frame}_n\text{>}$

But we are interested in the idiosyncratic, lexical usage of the verbs. The H-L PCFG lexicalisation of the grammar rules with their verb heads leads to a lexicalised distribution over frame types.

freq_1 C-<type>$^{[verb]}$ $\rightarrow$ S-<type>.$\text{<frame}_1\text{>}$
freq_2 C-<type>$^{[verb]}$ $\rightarrow$ S-<type>.$\text{<frame}_2\text{>}$
$\text{freq}_{...}$ C-<type>$^{[verb]}$ $\rightarrow$ S-<type>.$\text{<frame}_{...}\text{>}$
freq_n C-<type>$^{[verb]}$ $\rightarrow$ S-<type>.$\text{<frame}_n\text{>}$

Generalising over the clause type, the combination of grammar rules and lexical head information provides distributions for each verb over its subcategorisation frame properties.

$\texttt{freq}_1$ $\texttt{C}^{[verb]} \rightarrow \texttt{S.<frame}_1\texttt{>}$
$\texttt{freq}_2$ $\texttt{C}^{[verb]} \rightarrow \texttt{S.<frame}_2\texttt{>}$
$\texttt{freq}_{...}$ $\texttt{C}^{[verb]} \rightarrow \texttt{S.<frame}_{...}\texttt{>}$
$\texttt{freq}_n$ $\texttt{C}^{[verb]} \rightarrow \texttt{S.<frame}_n\texttt{>}$

An example of such a purely syntactic subcategorisation distribution is given in Table 3.16. The table lists the 38 subcategorisation frame types in the grammar sorted by the joint frequency with the verb *glauben* 'to think, to believe'. In this example as well as in all following examples on frequency extraction from the grammar, the reader might wonder why the frequencies are real values and not necessarily integers. This has to do with the training algorithm which splits a frequency of 1 for each sentence in the corpus over all ambiguous parses. Therefore, rule and lexical parameters might be assigned a fraction of 1.

In addition to a purely syntactic definition of subcategorisation frames, the grammar provides detailed information about the types of argument PPs within the frames. For each of the prepositional phrase frame types in the grammar (`np, nap, ndp, npr, xp`), the joint frequency of a verb and the PP frame is distributed over the prepositional phrases, according to their frequencies in the corpus. For example, Table 3.17 illustrates the subcategorisation for *reden* 'to talk' and the frame type `np` whose total joint frequency is 1,121.35.

3.4.2 Selectional Preferences

The grammar provides selectional preference information on a fine-grained level: it specifies the possible argument realisations in form of lexical heads, with reference to a specific verb-frame-slot combination. I.e. the grammar provides frequencies for heads for each verb and each frame type and each argument slot of the frame type. The verb-argument frequencies are regarded as a particular strength of the statistical model, since the relationship between verb and selected subcategorised head refers to fine-grained frame roles. For illustration purposes, Table 3.18 lists nominal argument heads for the verb *verfolgen* 'to follow' in the accusative NP slot of the transitive frame type `na` (the relevant frame slot is underlined), and Table 3.19 lists nominal argument heads for the verb *reden* 'to talk' in the PP slot of the transitive frame type `np:Akk.über`. The examples are ordered by the noun frequencies. For presentation reasons, I set a frequency cut-off.

3.4.3 Related Work on H-L PCFGs

There is a large amount of work on the automatic induction of lexical information. In this section, I therefore concentrate on the description of related work within the framework of H-L PCFGs.

With reference to my own work, Schulte im Walde (2002b) presents a large-scale computational subcategorisation lexicon for 14,229 German verbs with a frequency between 1 and 255,676.

The lexicon is based on the subcategorisation frame acquisition as illustrated in Section 3.4.1. Since the subcategorisation frames represent the core part of the verb description in this thesis, the lexicon is described in more detail and evaluated against manual dictionary definitions in Section 3.5. The section also describes related work on subcategorisation acquisition in more detail.

Schulte im Walde (2003a) presents a database of collocations for German verbs and nouns. The collocations are induced from the statistical grammar model. Concerning verbs, the database concentrates on subcategorisation properties and verb-noun collocations with regard to their specific subcategorisation relation (i.e. the representation of selectional preferences); concerning nouns, the database contains adjectival and genitive noun phrase modifiers, as well as their verbal subcategorisation. As a special case of noun-noun collocations, a list of 23,227 German proper name tuples is presented. All collocation types are combined by a perl script which can be queried by the lexicographic user in order to extract relevant co-occurrence information on a specific lexical item. The database is ready to be used for lexicographic research and exploitation.

Zinsmeister and Heid (2002, 2003b) utilise the same statistical grammar framework for lexical induction: Zinsmeister and Heid (2002) perform an extraction of noun-verb collocations, whose results represent the basis for comparing the collocational preferences of compound nouns with those of the respective base nouns. The insights obtained in this way are used to improve the lexicon of the statistical parser. Zinsmeister and Heid (2003b) present an approach for German collocations with collocation triples: Five different formation types of adjectives, nouns and verbs are extracted from the most probable parses of German newspaper sentences. The collocation candidates are determined automatically and then manually investigated for lexicographic use.

Frame Type	Freq
ns-dass	1,928.52
ns-2	1,887.97
np	686.76
n	608.05
na	555.23
ni	346.10
nd	234.09
nad	160.45
nds-2	69.76
nai	61.67
ns-w	59.31
nas-w	46.99
nap	40.99
nr	31.37
nar	30.10
nrs-2	26.99
ndp	24.56
nas-dass	23.58
nas-2	19.41
npr	18.00
nds-dass	17.45
ndi	11.08
nrs-w	2.00
nrs-dass	2.00
ndr	2.00
nir	1.84
nds-w	1.68
xd	1.14
ns-ob	1.00
nas-ob	1.00
x	0.00
xa	0.00
xp	0.00
xr	0.00
xs-dass	0.00
nds-ob	0.00
nrs-ob	0.00
k	0.00

Table 3.16: Subcategorisation frame distribution for *glauben*

Refined Frame Type	Freq
np:Akk.über	479.97
np:Dat.von	463.42
np:Dat.mit	279.76
np:Dat.in	81.35
np:Nom.vgl	13.59
np:Dat.bei	13.10
np:Dat.über	13.05
np:Dat.an	12.06
np:Akk.für	9.63
np:Dat.nach	8.49
np:Dat.zu	7.20
np:Dat.vor	6.75
np:Akk.in	5.86
np:Dat.aus	4.78
np:Gen.statt	4.70
np:Dat.auf	4.34
np:Dat.unter	3.77
np:Akk.vgl	3.55
np:Akk.ohne	3.05
np:Dat.hinter	3.00
np:Dat.seit	2.21
np:Dat.neben	2.20
np:Dat.wegen	2.13
np:Akk.gegen	2.13
np:Akk.an	1.98
np:Gen.wegen	1.77
np:Akk.um	1.66
np:Akk.bis	1.15
np:Akk.ab	1.13
np:Dat.laut	1.00
np:Gen.hinsichtlich	1.00
np:Gen.während	0.95
np:Dat.zwischen	0.92
np:Akk.durch	0.75

Table 3.17: Refined np distribution for *reden*

Noun		Freq
Ziel	‘goal’	86.30
Strategie	‘strategy’	27.27
Politik	‘policy’	25.30
Interesse	‘interest’	21.50
Konzept	‘concept’	16.84
Entwicklung	‘development’	15.70
Kurs	‘direction’	13.96
Spiel	‘game’	12.26
Plan	‘plan’	10.99
Spur	‘trace’	10.91
Programm	‘program’	8.96
Weg	‘way’	8.70
Projekt	‘project’	8.61
Prozeß	‘process’	7.60
Zweck	‘purpose’	7.01
Tat	‘action’	6.64
Täter	‘suspect’	6.09
Setzung	‘settlement’	6.03
Linie	‘line’	6.00
Spektakel	‘spectacle’	6.00
Fall	‘case’	5.74
Prinzip	‘principle’	5.27
Ansatz	‘approach’	5.00
Verhandlung	‘negotiation’	4.98
Thema	‘topic’	4.97
Kampf	‘combat’	4.85
Absicht	‘purpose’	4.84
Debatte	‘debate’	4.47
Karriere	‘career’	4.00
Diskussion	‘discussion’	3.95
Zeug	‘stuff’	3.89
Gruppe	‘group’	3.68
Sieg	‘victory’	3.00
Räuber	‘robber’	3.00
Ankunft	‘arrival’	3.00
Sache	‘thing’	2.99
Bericht	‘report’	2.98
Idee	‘idea’	2.96
Traum	‘dream’	2.84
Streit	‘argument’	2.72

Table 3.18: Nominal arguments for *verfolgen* in na

Noun		Freq
Geld	‘money’	19.27
Politik	‘politics’	13.53
Problem	‘problem’	13.32
Thema	‘topic’	9.57
Inhalt	‘content’	8.74
Koalition	‘coalition’	5.82
Ding	‘thing’	5.37
Freiheit	‘freedom’	5.32
Kunst	‘art’	4.96
Film	‘movie’	4.79
Möglichkeit	‘possibility’	4.66
Tod	‘death’	3.98
Perspektive	‘perspective’	3.95
Konsequenz	‘consequence’	3.90
Sache	‘thing’	3.73
Detail	‘detail’	3.65
Umfang	‘extent’	3.00
Angst	‘fear’	3.00
Gefühl	‘feeling’	2.99
Besetzung	‘occupation’	2.99
Ball	‘ball’	2.96
Sex	‘sex’	2.02
Sekte	‘sect’	2.00
Islam	‘Islam’	2.00
Fehler	‘mistake’	2.00
Erlebnis	‘experience’	2.00
Abteilung	‘department’	2.00
Demokratie	‘democracy’	1.98
Verwaltung	‘administration’	1.97
Beziehung	‘relationship’	1.97
Angelegenheit	‘issue’	1.97
Gewalt	‘force’	1.89
Erhöhung	‘increase’	1.82
Zölle	‘customs’	1.00
Vorsitz	‘chair’	1.00
Virus	‘virus’	1.00
Ted	‘Ted’	1.00
Sitte	‘custom’	1.00
Ressource	‘resource’	1.00
Notwendigkeit	‘necessity’	1.00

Table 3.19: Nominal arguments for *reden über*$_{Akk}$ ‘to talk about’

3.5 Grammar Evaluation

This final part of the grammar chapter describes an evaluation performed on the core of the grammar, its subcategorisation frames. I evaluated the verb subcategorisation frames which are learned in the statistical grammar framework against manual definitions in the German dictionary *Duden – Das Stilwörterbuch*. The work was performed in collaboration with *Bibliographisches Institut & F. A. Brockhaus AG* who provided a machine readable version of the dictionary. The evaluation is published by Schulte im Walde (2002a).

Section 3.5.1 describes the definition of verb subcategorisation frames (i) in the large-scale computational subcategorisation lexicon based on the statistical grammar model and (ii) in the manual dictionary *Duden*. In Section 3.5.2 the evaluation experiment is performed, Section 3.5.3 contains an interpretation of the experiment results, and Section 3.5.4 compares them with related work on English and German subcategorisation induction.

3.5.1 Subcategorisation Lexica for Verbs

Learning a Verb Subcategorisation Lexicon

Schulte im Walde (2002b) presents a large-scale computational subcategorisation lexicon. The lexicon is based on the empirical subcategorisation frame acquisition as illustrated in Section 3.4.1. The induction of the subcategorisation lexicon uses the trained frequency distributions over frame types for each verb. The frequency values are manipulated by squaring them, in order to achieve a more clear-cut threshold for lexical subcategorisation. The manipulated values are normalised and a cut-off of 1% defines those frames which are part of the lexical verb entry.

The manipulation is no high mathematical transformation, but it has the following impact on the frequency distributions. Assume verb v_1 has a frequency of 50 for the frame f_a and a frequency of 10 for frame f_b; verb v_2 has a frequency of 500 for the frame f_a and a frequency of 10 for frame f_b. If we set the cut-off to a frequency of 10, for example, then for both verbs both frames f_a and f_b are listed in the subcategorisation lexicon (but note that f_b is empirically less confirmed for v_2 than for v_1). If we set the cut-off to a frequency of 50, for example, then v_1 would have no frame listed at all. It is difficult to find a reliable cut-off. If we based the decision on the respective probability values p_a and p_b ($\langle v_1, p_a \rangle = 0.83$, $\langle v_1, p_b \rangle = 0.17$, $\langle v_2, p_a \rangle = 0.98$, $\langle v_2, p_b \rangle = 0.02$) it is easier to find a reliable cut-off, but still difficult for a large number of examples. But if we first square the frequencies ($\langle v_1, f'_a \rangle = 250$, $\langle v_1, f'_b \rangle = 100$, $\langle v_2, f'_a \rangle = 250,000$, $\langle v_2, f'_b \rangle = 100$), the respective probability values ($\langle v_1, p'_a \rangle = 0.71$, $\langle v_1, p'_b \rangle = 0.29$, $\langle v_2, p'_a \rangle = 0.9996$, $\langle v_2, p'_b \rangle = 0.0004$) are stretched, and it is not as difficult as before to find a suitable cut-off.

Tables 3.20 and 3.21 cite the (original and manipulated) frequencies and probabilities for the verbs *befreien* 'to free' and *zehren* 'to live on, to wear down' and mark the demarcation of lexicon-relevant frames by an extra line in the rows on manipulated numbers. The set of marked frames corresponds to the lexical subcategorisation for the respective verb.

Frame	Freq (orig)	Prob (orig)	Freq (mani)	Prob (mani)
na	310.50	0.43313	96,410.25	0.74293
nr	137.14	0.19130	18,807.38	0.14493
nap	95.10	0.13266	9,044.01	0.06969
n	59.04	0.08236	3,485.72	0.02686
nad	29.62	0.04132	877.34	0.00676
npr	23.27	0.03246	541.49	0.00417
np	15.04	0.02098	226.20	0.00174
nd	11.88	0.01657	141.13	0.00109
ndr	11.87	0.01656	140.90	0.00109
ns-2	7.46	0.01041	55.65	0.00043
nar	3.00	0.00418	9.00	0.00007
nrs-2	3.00	0.00418	9.00	0.00007
nds-2	2.94	0.00418	8.64	0.00007
nai	2.01	0.00280	4.04	0.00003
nir	2.00	0.00279	4.00	0.00003
ni	2.00	0.00279	4.00	0.00003
nas-2	1.00	0.00139	1.00	0.00001

Lexical subcategorisation: { n, na, nr, nap }

Table 3.20: Lexical subcategorisation for *befreien*

Frame	Freq (orig)	Prob (orig)	Freq (mani)	Prob (mani)
n	43.20	0.47110	1866.24	0.54826
np	38.71	0.42214	1498.46	0.44022
na	4.79	0.05224	22.94	0.00674
nap	3.87	0.04220	14.98	0.00440
nd	1.13	0.01232	1.28	0.00038

Lexical subcategorisation: { n, np }

Table 3.21: Lexical subcategorisation for *zehren*

A refined version of subcategorisation frames includes the specific kinds of prepositional phrases for PP-arguments. The frame frequency values and the PP frequency values are also manipulated by squaring them, and the manipulated values are normalised. The product of frame probability and PP probability is calculated, and a cut-off of 20% defines those PP frame types which are part of the lexical verb entry. The resulting lexical subcategorisation for *befreien* would be { n, na, nr, nap:Dat.von, nap:Dat.aus }, for *zehren* { n, np:Dat.von, np:Dat.an }.

I collected frames for all lexical items that were identified as verbs in the training corpus at least once, according to the definitions in the German morphological analyser AMOR underlying the grammar terminals. The resulting verb lexicon on subcategorisation frames contains 14,229 German verbs with a frequency between 1 and 255,676. Examples for lexical entries in the subcategorisation are given by Table 3.22 on the purely syntactic frame types, and by Table 3.23 on the PP-refined frame types.

Lexicon Entry			
Verb		Freq	Subcategorisation
aufregen	'to get excited'	135	na, nr
beauftragen	'to order', 'to charge'	230	na, nap, nai
bezweifeln	'to doubt'	301	na, ns-dass, ns-ob
bleiben	'to stay', 'to remain'	20,082	n, k
brechen	'to break'	786	n, na, nad, nar
entziehen	'to take away'	410	nad, ndr
irren	'to be mistaken'	276	n, nr
mangeln	'to lack'	438	x, xd, xp
scheinen	'to shine', 'to seem'	4,917	n, ni
sträuben	'to resist'	86	nr, npr

Table 3.22: Examples for purely syntactic lexical subcategorisation entries

Lexicon Entry			
Verb		Freq	Subcategorisation
beauftragen	'to order', 'to charge'	230	na, nap:Dat.mit, nai
denken	'to think'	3,293	n, na, np:Akk.an, ns-2
enden	'to end'	1,900	n, np:Dat.mit
ernennen	'to appoint'	277	na, nap:Dat.zu
fahnden	'to search'	163	np:Dat.nach
klammern	'to cling to'	49	npr:Akk.an
schätzen	'to estimate'	1,357	na, nap:Akk.auf
stapeln	'to pile up'	137	nr, npr:Dat.auf, npr:Dat.in
sträuben	'to resist'	86	nr, npr:Akk.gegen
tarnen	'to camouflage'	32	na, nr, npr:Nom.vgl

Table 3.23: Examples for PP-refined lexical subcategorisation entries

Manual Definition of Subcategorisation Frames in Dictionary *Duden*

The German dictionary *Duden – Das Stilwörterbuch* (Dudenredaktion, 2001) describes the stylistic usage of words in sentences, such as their syntactic embedding, example sentences, and idiomatic expressions. Part of the lexical verb entries are frame-like syntactic descriptions, such as `<jmdn. befreien>` 'to free somebody' with the direct object indicated by the accusative case, or `<von etw. zehren>` 'to live on something$_{Dat}$'.

Duden does not contain explicit subcategorisation frames, since it is not meant to be a subcategorisation lexicon. But it does contain 'grammatical information' for the description of the stylistic usage of verbs; therefore, the *Duden* entries implicitly contain subcategorisation, which enables us to infer frame definitions.

Alternations in verb meaning are marked by a semantic numbering `SEM`*X*`-ID` and accompanied by the respective subcategorisation requirements (`GR` provides the subcategorisation, `DEF` provides a semantic description of the respective verb usage, and `TEXT` under `BSP` provides examples for selectional preferences). For example, the lexical verb entry for *zehren* in Figure 3.18 lists the following lexical semantic verb entries:

1. `<von etw. zehren>` 'to live on something'
2. 'to drain somebody of his energy'
   a) no frame which implicitly refers to an intransitive usage
   b) `<an jmdm., etw. zehren>`

Idiosyncrasies in the manual frame definitions lead to a total of 1,221 different subcategorisation frames in *Duden*:

- Subcategorised elements might be referred to either by a specific category or by a general item, for example `irgendwie` 'somehow' comprises the subcategorisation of any prepositional phrase:
  <irgendwie>
  But prepositional phrases might also be made explicit:
  <für etw.>
  A similar behaviour is exhibited for the *Duden* expressions `irgendwo` 'somewhere', `irgendwohin` 'to some place', `irgendwoher` 'from some place', `irgendwann` 'some time', `mit Umstandsangabe` 'under some circumstances'.

- Identical frame definitions differ in their degree of explicitness, for example
  <[gegen jmdn., etw. (Akk.)]>
  <[gegen jmdn., etw.]>
  both refer to the potential (indicated by '[]') subcategorisation of a prepositional phrase with accusative case and head *gegen* 'against'. The former frame explicitly refers to the accusative case, the latter implicitly needs the case because the preposition demands accusative case.

- In some cases, *Duden* distinguishes between animate and non-animate selectional restrictions, for example
  <etw. auf etw. (Akk.)>
  <etw. auf jmdn.>
  <etw. auf jmdn., etw.>
  <etw. auf ein Tier>
  <jmdn. auf etw. (Akk.)>
  <jmdn. auf jmdn.>
  <jmdn. auf jmdn., etw.>
  <jmdn. auf ein Tier>
  <ein Tier auf etw. (Akk.)>
  <ein Tier auf jmdn.>
  <ein Tier auf jmdn., etw.>
  all refer to a transitive frame with obligatory prepositional phrase `Akk.auf`.

- Syntactic comments in *Duden* might refer to a change in the subcategorisation with reference to another frame, but the modified subcategorisation frame is not explicitly provided. For example, <auch mit Akk.> refers to a modification of a frame which allows the verb to add an accusative noun phrase.

Correcting and reducing the idiosyncratic frames to their common information concerning our needs results in 65 subcategorisation frames without explicit prepositional phrase definitions and 222 subcategorisation frames including them.

The lexicon is implemented in `SGML`. I defined a *Document Type Definition (DTD)* which formally describes the structure of the verb entries and extracted manually defined subcategorisation frames for 3,658 verbs from the *Duden*.

```
<D2>

  <SEM1 SEM1-ID="1">
    <DEFPHR>
      <GR><von etw. zehren> </GR>
      <DEF>etw. aufbrauchen: </DEF>
      <BSP>
        <TEXT>von den Vorräten, von seinen Ersparnissen zehren; </TEXT>
      </BSP>
     </DEFPHR>
  </SEM1>

  <SEM1 SEM1-ID="2">

  <SEM2 SEM2-ID="a">
    <DEFPHR>
      <DEF>schwächen: </DEF>
      <BSP>
        <TEXT>das Fieber, die Seeluft, die See zehrt; </TEXT>
        <TEXT>eine zehrende Krankheit; </TEXT>
      </BSP>
    </DEFPHR>
  </SEM2>

  <SEM2 SEM2-ID="b">
    <DEFPHR>
      <GR><an jmdm., etw. zehren> </GR>
      <DEF>jmdm., etw. sehr zusetzen: </DEF>
      <BSP>
        <TEXT>das Fieber, die Krankheit zehrte an seinen Kräften; </TEXT>
        <TEXT>der Stress zehrt an ihrer Gesundheit; </TEXT>
        <TEXT>die Sorge, der Kummer, die Ungewissheit hat sehr an ihr,
              an ihren Nerven gezehrt. </TEXT>
      </BSP>
    </DEFPHR>
  </SEM2>

  </SEM1>

</D2>
```

Figure 3.18: *Duden* lexical entry for *zehren*

3.5.2 Evaluation of Subcategorisation Frames

Frame Mapping Preceding the actual experiment I defined a deterministic mapping from the *Duden* frame definitions onto my subcategorisation frame style, e.g. the ditransitive frame definition `<jmdm. etw.>` would be mapped to `nad`, and `<bei jmdm. etw.>` would be mapped to `nap` without and `nap:Dat.bei` with explicit prepositional phrase definition. 38 *Duden* frames do not match anything in my frame repertoire (mostly rare frames such as `nag` *Er beschuldigt ihn des Mordes* 'He accuses him of the murder', or frame types with more than three arguments); 5 of my frame types do not appear in the *Duden* (copula constructions and some frames including finite clause arguments such as `nds-2`).

Evaluation Measures For the evaluation of the learned subcategorisation frames, the manual *Duden* frame definitions are considered as the gold standard. I calculated precision and recall values on the following basis:

$$recall = \frac{tp}{tp + fn} \tag{3.4}$$

$$precision = \frac{tp}{tp + fp} \tag{3.5}$$

tp (true positives) refer to those subcategorisation frames where learned and manual definitions agree, fn (false negatives) to the *Duden* frames not extracted automatically, and fp (false positives) to those automatically extracted frames not defined by *Duden*.

Major importance is given to the f-score which considers recall and precision as equally relevant and therefore balances the previous measures:

$$f - score = \frac{2 * recall * precision}{recall + precision} \tag{3.6}$$

Experiments The evaluation experiment has three conditions.

I All frame types are taken into consideration. In case of a prepositional phrase argument in the frame, the PP is included, but the refined definition is ignored, e.g. the frame including one obligatory prepositional phrase is referred to by `np` (nominative noun phrase plus prepositional phrase).

II All frame types are taken into consideration. In case of a prepositional phrase argument in the frame, the refined definition is included, e.g. the frame including one obligatory prepositional phrase (cf. I) is referred to by `np:Akk.für` for a prepositional phrase with head *für* and the accusative case, `np:Dat.bei` for a prepositional phrase with head *bei* and the dative case, etc.

III Prepositional phrases are excluded from subcategorisation, i.e. frames including a p are mapped to the same frame type without that argument. In this way, a decision between prepositional phrase arguments and adjuncts is avoided.

Assuming that predictions concerning the rarest events (verbs with a low frequency) and those concerning the most frequent verbs (with increasing tendency towards polysemy) are rather unreliable, I performed the experiments on those 3,090 verbs in the *Duden* lexicon with a frequency between 10 and 2,000 in the corpus. See Table 3.24 for a distribution over frequency ranges for all 3,658 verbs with frequencies between 1 and 101,003. The horizontal lines mark the restricted verb set.

Freq			Verbs
1	-	5	162
5	-	10	289
10	-	20	478
20	-	50	690
50	-	100	581
100	-	200	496
200	-	500	459
500	-	1000	251
1000	-	2000	135
2000	-	5000	80
5000	-	10000	24
	>	10000	13

Table 3.24: Frequencies of *Duden* verbs in training corpus

Baseline As baseline for the experiments, I assigned the most frequent frame types n (intransitive frame) and na (transitive frame) as default to each verb.

Results The experimental results are displayed in Table 3.25.

Experiment	Recall		Precision		F-Score	
	Baseline	Result	Baseline	Result	Baseline	Result
I	49.57%	63.91%	54.01%	60.76%	51.70%	62.30%
II	45.58%	50.83%	54.01%	65.52%	49.44%	57.24%
III	63.92%	69.74%	59.06%	74.53%	61.40%	72.05%

Table 3.25: Evaluation of subcategorisation frames

Concerning the f-score, I reach a gain of 10% compared to the baseline for experiment I: evaluating all frame definitions in the induced lexicon including prepositional phrases results in 62.30% f-score performance. Complicating the task by including prepositional phrase definitions into the frame types (experiment II), I reach 57.24% f-score performance, 8% above the baseline. Completely disregarding the prepositional phrases in the subcategorisation frames (experiment III) results in 72.05% f-score performance, 10% above the baseline.

The differences both in the absolute f-score values and the difference to the respective baseline values correspond to the difficulty and potential of the tasks. Disregarding the prepositional phrases completely (experiment III) is the easiest task and therefore reaches the highest f-score. But the baseline frames n and na represent 50% of all frames used in the *Duden* lexicon, so the potential for improving the baseline is small. Compared to experiment III, experiment I is a more difficult task, because the prepositional phrases are taken into account as well. But I reach a gain in f-score of more than 10%, so the learned frames can improve the baseline decisions. Experiment II shows that defining prepositional phrases in verb subcategorisation is a more complicated task. Still, I improve the baseline results by 8%.

3.5.3 Lexicon Investigation

Section 3.5.2 presented the figures of merit for verb subcategorisation frames which are learned in the statistical grammar framework against the manual verb descriptions in the German dictionary *Duden*. The current section discusses advantages and shortcomings of the verb subcategorisation lexica concerning the selection of verbs and detail of frame types.

The verb entries in the automatic and manual subcategorisation lexica are examined: the respective frames are compared, against each other as well as against verb entries in Helbig and Schenkel (1969) (henceforth: H/S) and corpus evidence in the German newspaper corpus *die tageszeitung (TAZ)*. In addition, I compare the set of frames in the two lexica, their intersection and differences. The result of the investigation is a description of strengths and deficiencies in the lexica.

Intransitive Verbs In the *Duden* dictionary, intransitive verb usage is difficult to extract, since it is defined only implicitly in the verb entry, such as for the verbs *glücken* ‘to succeed’, *langen* ‘to suffice’, *verzweifeln* ‘to despair’. In addition, *Duden* defines the intransitive frame for verbs which can be used intransitively in exclamations, such as *Der kann aber wetzen!* ‘Wow, he can dash!’. But the exclamatory usage is not sufficient evidence for intransitive usage. The induced lexicon, on the other hand, tends to overgenerate the intransitive usage of verbs, mainly because of parsing mistakes. Still, the intersection of intransitive frames in both lexica reaches a recall of 77.19% and a precision of 66.11%,

Transitive Verbs The usage of transitive verbs in the lexica is the most frequent occurrence and at the same time the most successfully learned frame type. *Duden* defines transitive frames for 2,513 verbs, the automatic process extracts 2,597 frames. An agreement in 2,215 cases corresponds to 88.14% recall and 85.29% precision.

Dative Constructions *Duden* verb entries are inconsistent concerning the free dative construction ('freier Dativ'). For example, the free dative is existing in the ditransitive usage for the verb *ablösen* 'to remove' (*Der Arzt löste ihm das Pflaster ab* 'The doctor removed him the plaster), but not for the verb *backen* 'to bake' (H/S: *Die Mutter backt ihm einen Kuchen* 'The mother baked him a cake'). The induced lexicon is rather unreliable on frames including dative noun phrases. Parsing mistakes tend to extract accusative constructions as dative and therefore wrongly emphasise the dative usage.

Prepositional Phrases In general, *Duden* properly distinguishes between prepositional phrase arguments (mentioned in subcategorisation) and adjuncts, but in some cases, *Duden* overemphasises certain PP-arguments in the verb frame definition, such as `Dat.mit` for the verbs *aufschließen* 'to unlock', *garnieren* 'to garnish', *nachkommen* 'to keep up', `Dat.von` for the verbs *abbröckeln* 'to crumble', *ausleihen* 'to borrow', *erbitten* 'to ask for', *säubern* 'to clean up', or `Akk.auf` for the verbs *abklopfen* 'to check the reliability', *ausüben* 'to practise', *festnageln* 'to tie down', *passen* 'to fit'.

In the induced lexicon, prepositional phrase arguments are overemphasised, i.e. PPs used as adjuncts are frequently inserted into the lexicon, e.g. for the verbs *arbeiten* 'to work', *demonstrieren* 'to demonstrate', *sterben* 'to die'. This mistake is mainly based on highly frequent prepositional phrase adjuncts, such as `Dat.in, Dat.an, Akk.in`. On the other hand, the induced lexicon does not recognise verb-specific prepositional phrase arguments in some cases, such as `Dat.mit` for the verbs *gleichstellen* 'to equate', *handeln* 'to act', *spielen* 'to play', or `Dat.von` for the verbs *abbringen* 'to dissuade', *fegen* 'to sweep', *genesen* 'to convalesce', *schwärmen* 'to romanticise'.

Comparing the frame definitions containing PPs in both lexica, the induced lexicon tends to define PP-adjuncts such as `Dat.in, Dat. an` as arguments and neglect PP-arguments; *Duden* distinguishes arguments and adjuncts more correctly, but tends to overemphasise PPs such as `Dat.mit` and `Dat.bei` as arguments. Still, there is agreement on the `np` frame with 59.69% recall and 49.88% precision, but the evaluation of `nap` with 45.95% recall, 25.89% precision and of `ndp` with 9.52% recall and 15.87% precision pinpoints main deficiencies in the frame agreement.

Reflexive Verbs *Duden* generously categorises verbs as reflexives; they appear whenever it is possible to use the respective verb with a reflexive pronoun. The procedure is valid for verbs such as *erwärmen* 'to heat', *lohnen* 'to be worth', *schämen* 'to feel ashamed', but not for verbs

such as *durchbringen* ‘to pull through’, *kühlen* ‘to cool’, *zwingen* ‘to force’. The automatic frame definitions, on the other hand, tend to neglect the reflexive usage of verbs and rather choose direct objects into the frames, such as for the verbs *ablösen* ‘to remove’, *erschießen* ‘to shoot’, *überschätzen* ‘to overestimate’. The lexicon tendencies are reflected by the `nr, nar, npr` frame frequencies: rather low recall values between 28.74% and 45.17%, and rather high precision values between 51.94% and 69.34% underline the differences.

Adjectival Phrases The definition of adjectival phrase arguments in the *Duden* is somewhat idiosyncratic, especially as demarcation to non-subcategorised adverbial phrases. For example, an adjectival phrase for the verb *scheinen* ‘to shine’ as in *Die Sonne schien hell* ‘The sun is bright’ is subcategorised, as well as for the verb *berühren* ‘to touch’ as in *Seine Worte haben uns tief berührt* ‘His words touched us deeply’. Concerning the induced lexicon, the grammar does not contain adjectival phrase arguments, so they could not be recognised, such as for the verbs *anmuten* ‘to seem’, *erscheinen* ‘to seem’, *verkaufen* ‘to sell’.

Subcategorisation of Clauses *Duden* shows shortcomings on the subcategorisation of non-finite and finite clauses; they rarely appear in the lexicon. Only 26 verbs (such as *anweisen* ‘to instruct’, *beschwören* ‘to swear’, *versprechen* ‘to promise’) subcategorise non-finite clauses, only five verbs (such as *sehen* ‘to see’, *wundern* ‘to wonder’) subcategorise finite clauses. Missing verbs for the subcategorisation of finite clauses are –among others– *ausschließen* ‘to rule out’, *sagen* ‘to say’, *vermuten* ‘to assume’, for the subcategorisation of non-finite clauses *hindern* ‘to prevent’, *verpflichten* ‘to commit’.

The automatic lexicon defines the subcategorisation of clauses more reliably. For example, the verbs *behaupten* ‘to state’, *nörgeln* ‘to grumble’ subcategorise verb second finite clauses, the verbs *aufpassen* ‘to pay attention’, *glauben* ‘to think’, *hoffen* ‘to hope’ subcategorise finite *dass*-clauses, the verb *bezweifeln* ‘to doubt’ subcategorises a finite *ob*-clause, the verbs *ahnen* ‘to guess’, *klarmachen* ‘to make clear’, *raffen* ‘to understand’ subcategorise indirect *wh*-questions, and the verbs *anleiten* ‘to instruct’, *beschuldigen* ‘to accuse’, *lehren* ‘to teach’ subcategorise non-finite clauses. Mistakes occur for indirect *wh*-questions which are confused with relative clauses, such as for the verbs *ausbaden* ‘to pay for’, *futtern* ‘to eat’.

General Frame Description *Duden* defines verb usage on various levels of detail, especially concerning prepositional phrases (cf. Section 2.2). For example, `irgendwie` ‘somehow’ in grammatical definitions means the usage of a prepositional phrase such as for the verb *lagern* ‘to store’ (*Medikamente müssen im Schrank lagern* ‘Drugs need to be stored in a cupboard’); `irgendwo` ‘somewhere’ means the usage of a locative prepositional phrase such as for the verb *lauern* ‘to lurk’ (*Der Libero lauert am Strafraum* ‘The sweeper lies in wait in the penalty area.’). In more restricted cases, the explicit prepositional phrase is given as in `<über etw. (Akk.)>` for the verb *verzweifeln* ‘to despair’ (*Man könnte verzweifeln über so viel Ignoranz* ‘One could despair about that ignorance’).

The grammatical definitions on various levels of detail are considered as a strength of *Duden* and generally favourable for users of a stylistic dictionary, but produce difficulties for automatic usage. For example, when including PP-definitions into the evaluation (experiment II), 10% of the *Duden* frames (PP-frames without explicit PP-definition, such as `np`) could never be guessed correctly, since the automatic lexicon includes the PPs explicitly.

There are frame types in *Duden* which do not exist in the automatic verb lexicon. This mainly concerns rare frames such as `nag, naa, xad` and frame types with more than three arguments such as `napr, ndpp`. This lexicon deficiency concerns about 4% of the total number of frames in the *Duden* lexicon.

Lexicon Coverage Compared to the automatic acquisition of verbs, *Duden* misses verbs in the dictionary: frequent verbs such as *einreisen* 'to enter', *finanzieren* 'to finance', *veranschaulichen* 'to illustrate', verbs adopted from English such as *dancen, outen, tunen*, vulgar verbs such as *anpöbeln* 'to abuse', *ankotzen* 'to make sick', *pissen* 'to piss', recent neologisms such as *digitalisieren* 'to digitalise', *klonen* 'to clone', and regional expressions such as *kicken* 'to kick', *latschen* 'to walk', *puhlen* 'to pick'.

The automatic acquisition of verbs covers a larger amount of verbs, containing 14,229 verb entries, including the missing examples above. Partly, mistaken verbs are included in the lexicon: verbs wrongly created by the morphology such as **angebieten, *dortdrohen, *einkommen*, verbs which obey the old, but not the reformed German spelling rules such as *autofahren* 'to drive a car', *danksagen* 'to thank', *spazierengehen* 'to stroll', and rare verbs, such as *?bürgermeistern, ?evangelisieren, ?fiktionalisieren, ?feuerwerken, ?käsen*.

Table 3.26 summarises the lexicon investigation. I blindly classified 184 frame assignments from fn and fp into correct and wrong. The result emphasises (i) unreliabilities for `n` and `nd` in both lexica, (ii) insecurities for reflexive and expletive usage in both lexica, (iii) strength of clause subcategorisation in the induced lexicon (the few assignments in *Duden* were all correct), (iv) strength of PP-assignment in the *Duden*, and (v) variability of PP-assignment in the induced lexicon.

Summary The lexicon investigation showed that

- in both lexica, the degree of reliability of verb subcategorisation information depends on the different frame types. If I tried different probability thresholds for different frame types, the accuracy of the subcategorisation information should improve once more.
- we need to distinguish between the different goals of the subcategorisation lexica: the induced lexicon explicitly refers to verb arguments which are (obligatorily) subcategorised by the verbs in the lexicon, whereas *Duden* is not intended to represent a subcategorisation lexicon but rather to describe the stylistic usage of the verbs and therefore to refer to possibly subcategorised verb arguments; in the latter case, there is no distinction between obligatory and possible verb complementation.

Frame Type	*Duden*: fn		Learned: fp	
	correct	wrong	correct	wrong
n	4	6	3	7
nd	2	8	0	10
nr, nar, ndr	5	5	3	7
x, xa, xd, xr	6	4	3	7
ni, nai, ndi			5	5
ns/nas/nds-dass			9	0
ns/nas/nds-2			9	1
np/nap/ndp/npr:Dat.mit	7	3	6	4
np/nap/ndp/npr:Dat.von	7	3	5	0
np/nap/ndp/npr:Dat.in	6	4	3	7
np/nap/ndp/npr:Dat.an	9	1	6	4

Table 3.26: Investigation of subcategorisation frames

- a manual lexicon suffers from the human potential of permanently establishing new words in the vocabulary; it is difficult to be up-to-date, and the learned lexical entries therefore hold a potential for adding to and improving manual verb definitions.

3.5.4 Related Work

Automatic induction of subcategorisation lexica has mainly been performed for English. Brent (1993) uses unlabelled corpus data and defines morpho-syntactic cues followed by a statistical filtering, to obtain a verb lexicon with six different frame types, without prepositional phrase refinement. Brent evaluates the learned subcategorisation frames against hand judgements and achieves an f-score of 73.85%. Manning (1993) also works on unlabelled corpus data and does not restrict the frame definitions. He applies a stochastic part-of-speech tagger, a finite state parser, and a statistical filtering process (following Brent). Evaluating 40 randomly selected verbs (out of 3,104) against *The Oxford Advanced Learner's Dictionary* (Hornby, 1985) results in an f-score of 58.20%. Briscoe and Carroll (1997) pre-define 160 frame types (including prepositional phrase definitions). They apply a tagger, lemmatiser and parser to unlabelled corpus data; from the parsed corpus they extract subcategorisation patterns, classify and evaluate them, in order to build the lexicon. The lexical definitions are evaluated against the Alvey NL Tools dictionary (Boguraev *et al.*, 1987) and the COMLEX Syntax dictionary (Grishman *et al.*, 1994) and achieve an f-score of 46.09%. The work in Carroll and Rooth (1998) is closest to ours, since they utilise the same statistical grammar framework for the induction of subcategorisation frames, but not including prepositional phrase definitions. Their evaluation for 200 randomly chosen verbs with a frequency greater than 500 against *The Oxford Advanced Learner's Dictionary* obtains an f-score of 76.95%.

For German, Eckle (1999) performs a semi-automatic acquisition of subcategorisation information for 6,305 verbs. She works on annotated corpus data and defines linguistic heuristics in the form of regular expression queries over the usage of 244 frame types including PP definitions. The extracted subcategorisation patterns are judged manually. Eckle performs an evaluation on 15 hand-chosen verbs; she does not cite explicit recall and precision values, except for a subset of subcategorisation frames. Wauschkuhn (1999) constructs a valency dictionary for 1,044 verbs with corpus frequency larger than 40. He extracts a maximum of 2,000 example sentences for each verb from annotated corpus data, and constructs a context-free grammar for partial parsing. The syntactic analyses provide valency patterns, which are grouped in order to extract the most frequent pattern combinations. The common part of the combinations define a distribution over 42 subcategorisation frame types for each verb. The evaluation of the lexicon is performed by hand judgement on seven verbs chosen from the corpus. Wauschkuhn achieves an f-score of 61.86%.

Comparing our subcategorisation induction with existing approaches for English, Brent (1993), Manning (1993) and Carroll and Rooth (1998) are more flexible than ours, since they do not require a pre-definition of frame types. But none of them includes the definition of prepositional phrases, which makes our approach the more fine-grained version. Brent (1993) outperforms our approach by an f-score of 73.85%, but the number of six frames is incomparable; Manning (1993) and Briscoe and Carroll (1997) both have f-scores below ours, even though the evaluations are performed on more restricted data. Carroll and Rooth (1998) reach the best f-score of 76.95% compared to 72.05% in our approach, but their evaluation is facilitated by restricting the frequency of the evaluated verbs to more than 500.

Concerning subcategorisation lexica for German, I have constructed the most independent approach I know of, since I do need either extensive annotation of corpora, nor restrict the frequencies of verbs in the lexicon. In addition, the approach is fully automatic after grammar definition and does not involve heuristics or manual corrections. Finally, the evaluation is not performed by hand judgement, but rather extensively on independent manual dictionary entries.

3.6 Summary

This chapter has described the implementation, training and lexical exploitation of the German statistical grammar model which serves as source for the German verb description at the syntax-semantic interface. I have introduced the theoretical background of the statistical grammar model and illustrated the manual implementation of the underlying German grammar. A training strategy has been developed which learns the large parameter space of the lexicalised grammar model. On the basis of various examples and related work, I illustrated the potential of the grammar model for an empirical lexical acquisition, not only for the purpose of verb clustering, but also for theoretical linguistic investigations and NLP applications such as lexicography and parsing improvement.

It is desirable but difficult to evaluate all of the acquired lexical information at the syntax-semantic interface. For a syntactic evaluation, manual resources such as the *Duden* dictionary are available, but few resources offer a manual definition of semantic information. So I concentrated on an evaluation of the subcategorisation frames as core part of the grammar model. The subcategorisation lexicon as based on the statistical framework has been evaluated against dictionary definitions and proven reliable: the lexical entries hold a potential for adding to and improving manual verb definitions. The evaluation results justify the utilisation of the subcategorisation frames as a valuable component for supporting NLP-tasks.