3.1 Allgemeine Angaben zum Teilprojekt D4

3.1.1 Titel:
Kurztitel: LexPar

Modular Lexicalization of Probabilistic Context-Free Grammars

3.1.2 Fachgebiete und Arbeitsrichtung:
Parsing, Statistical Methods, Lexicalized Probabilistic Context-Free Grammars

3.1.3 Leiter/in:
Schmid, Helmut

3.2 Zusammenfassung

Kurzzusammenfassung
This project aims to develop and implement a statistical disambiguation method for syntactic analyses (parse trees) which is based on lexicalized probabilistic context-free grammars (PCFGs). Unlike previous approaches, the statistical model employed in this project is modular and comprises two components, a standard unlexicalized PCFG which is trained on a treebank (a manually created collection of parse trees), and a separate set of parameters encoding lexical dependencies. In order to overcome sparse data problems, the lexical parameters will also be learned from large unparsed corpora using a bootstrapping approach. We are planning to train parsers for English, German and possibly other languages where a treebank is available. We expect that the model will also be useful for lexical research.

Langzusammenfassung
This project aims to develop and implement a statistical disambiguation method for syntactic analyses (parse trees) using lexicalized probabilistic context-free grammars (PCFGs). On the basis of an existing probabilistic treebank grammar parser for English, this project will develop lexicalized treebank parsers for German and English. The lexicalization component of our statistical model compensates for the violation of independence assumptions in the unlexicalized model. Unlexicalized PCFGs require that the probability of a word depends only on the part of speech and not on any other information from the sentence context. Our lexicalized PCFG weakens this independence assumption: The first version of the model permits that a word depends also on its lexical governor, which accounts for the fact that the noun “flower”, for instance, is more likely to be the lexical head of an object NP if the NP is governed by the verb “water” than if it is governed by the verb “read”, for example. The model includes lexical parameters for all word-governor pairs which violate the independence assumption of the unlexicalized model significantly, i.e. whose frequency is much higher than the unlexicalized model predicts. For the other word-governor pairs, the unlexicalized model is sufficiently accurate and requires no correction. This approach
keeps the number of parameters small and avoids sparse data problems because the word-governor pairs for which lexical parameters are estimated are frequent enough to allow reliable parameter estimation even without smoothing.

The project will investigate another lexicalization strategy where the lexical parameters represent word/argument-set dependencies instead of word/governor dependencies. Unlike the first approach, this model allows dependencies between the arguments of a word and is therefore able to assign a higher probability to the phrase the book in the context ... reads like a detective story than in the context ... reads the newspaper. This model addresses the sparse data problem by (i) clustering words into overlapping word classes and by (ii) using the general-purpose taxonomy WordNet (Miller et al., 1993) to generalize arguments to concepts. The model will be trained with the Expectation-Maximization algorithm. The training data consists of word argument tuples which have been extracted from corpora. The Minimum-Description-Length principle Rissanen (1978) will be applied to find the appropriate level of generalization in the Wordnet hierarchy for each argument slot. We anticipate that this model will also be useful for lexical research because it is able to identify word readings, to group them into word classes, to determine the subcategorization frames of these word classes and to identify the selectional restrictions of the argument slots in the form of WordNet concepts.

The unlexicalized part of the statistical disambiguation model consists of a PCFG. The PCFG will be extracted from a treebank which was automatically augmented with additional features in order to improve the parsing performance. The rule probabilities will be estimated from the treebank frequencies. The lexical parameters, on the other hand, will be learned from treebank data as well as from additional data obtained from automatically parsed corpora. We expect that the treebank data will not reveal all relevant lexical dependencies and that the additional information obtained from automatically parsed corpora will be beneficial even if it is less accurate due to parsing errors. Furthermore, the unsupervised training could be a way to adapt the parser to new genres without building a new treebank.

In summary, the project will investigate (i) how well the proposed lexicalization methods work in terms of parsing accuracy, (ii) whether unsupervised training on unlabeled data improves the performance (iii) to what extent generic lexical resources like WordNet are useful for statistical parse tree disambiguation (iv) and to what extent the second model is able to extract lexical information like word readings, word classes, subcategorization properties and selectional restrictions.

Deutsche Zusammenfassung

nicht erfasst. Zwei unterschiedliche Lexikalisierungsansätze werden verfolgt. In Modell 1 werden nur Abhängigkeiten zwischen einem Wort und seinem Regens betrachtet. Für jedes Wort-Regens-Paar, das signifikant häufiger auftritt, als auf Grund der Häufigkeiten der Einzelwörter zu erwarten wäre, wird ein Parameter gespeichert, der angibt, um welchen Faktor die Häufigkeit erhöht ist.


3.3 Ausgangssituation des Teilprojekts

3.3.1 Stand der Forschung

PCFG parsers have been extensively studied in the past ten years and their accuracy has gradually improved to over 90% (f-score) correctly identified constituents. PCFG parsers usually extract a context-free grammar and rule probabilities from a treebank – although there are also PCFG parsers which use a manually written grammar (see e.g. Carroll and Rooth, 1998). Simple unlexicalized PCFG parsers which read the grammar directly from a treebank (obtaining grammar rules like $S \rightarrow \text{NP \ VP}$ and $\text{VP} \rightarrow \text{VBZ \ NP}$) achieve high coverage – almost all sentences receive at least one parse tree – but the accuracy of the most likely parse (Viterbi parse) selected by the PCFG parser is rather low with only about 70% of the constituents being correct according to the f-score measure. The low accuracy is due to the limited amount of information conveyed by the treebank categories. The Penn treebank (Marcus et al., 1993), for instance, assigns the same label VP to verb phrases headed by finite verbs, infinitives, gerunds or past participles and the same label S to finite clauses, infinitive clauses and so on.

**Improved Unlexicalized PCFG Parsing**  In order to obtain better parsing results, the strong independence assumptions of the simple PCFG parser, namely that the structure of a constituent only depends on its treebank category, have to be weakened. In other
words, the information represented in the constituent labels has to be augmented. This can be done in a number of ways. The annotation of the treebank constituents with the part-of-speech (POS) tag of its lexical head (Eisner, 1996; Collins, 1997; Charniak, 2000), for instance, encodes morphosyntactic information and allows the parser to distinguish between finite VPs (VP/VBZ, VP/VBP, VP/VBD), infinitive VPs (VP/VB), gerund VPs (VP/VBG), etc. Another well-known extension of the basic PCFG approach is parent annotation (Johnson, 1998). Adding information about the parent category splits e.g. NPs into subject NPs (which are dominated by S nodes) and object NPs (which are dominated by VP nodes).

**Markovization** Treebanks like the English Penn treebank (Marcus et al., 1993) or the German Negra (Skut et al., 1998) or Tiger treebank often have a rather flat structure. An example is the phrase *other small apparel makers, button suppliers, trucking firms and fabric houses* from the Penn treebank which is analyzed as a noun phrase with the following sequence of daughter nodes: JJ JJ NN NNS , NN NNS , NN NNS CC NN NNS.¹ Grammars extracted from treebanks with such a flat constituent structure contain many low-frequency rules with long right-hand-sides. Many other possible rules like the rule NP → DT JJ NN NNS , NN NNS , NN NNS CC NN NNS which is just as likely as the noun phrase pattern in the above example, are missing in the treebank, but are needed to parse certain sentences. This sparse data problem has prompted the development of a new class of statistical parsers (Collins, 1997; Charniak, 2000) where the expansion of a non-terminal is determined by a Markov process rather than a fixed set of grammar rules. Klein and Manning (2003) achieved a similar effect by a grammar transformation called markovization. Instead of expanding an NP directly to, say, DT JJ JJ NN, the markovized grammar first expands² NP to the new auxiliary symbol ⟨NP,NN⟩ plus NN. ⟨NP,NN⟩ is further expanded to ⟨NP,JJ⟩ plus JJ. ⟨NP,JJ⟩ is expanded to ⟨NP,JJ⟩ JJ and the second ⟨NP,JJ⟩ is expanded to DT JJ in order to generate the same sequence DT JJ JJ NN. The advantage of the markovization is that the sequences DT JJ JJ JJ NN and DT JJ JJ JJ JJ NN etc. can be parsed with the same rules.

**Unknown Words** Unknown words pose a special problem to parsers. In order to process them, a statistical parser has to guess the set of possible parts of speech (POS). A simple strategy is the following (see Collins, 1997): All words appearing less than e.g. 5 times in the treebank are replaced by the special token UNK. The statistical parser which was trained on the modified treebank also replaces unknown words (words appearing less than 5 times in the training data) in the input with the token UNK and continues as usual. A better strategy is to partition unknown words into classes based on features like capitalization, prefixes, suffixes, inclusion of digits and other special characters and to estimate the POS probabilities of each word class from the POS frequencies of the known words belonging to the same word class. Klein and Manning (2003) also use these word class probabilities to smooth the POS probabilities of known words.

¹The meaning of these Penn treebank categories is as follows: JJ is an adjective with positive degree, NN is a singular noun, NNS is a plural noun, CC is a coordinating conjunction, DT is a determiner and “,” is the part-of-speech tag assigned to commas.

²The actual markovization in (Klein and Manning, 2003) is more complex because they also encode the head daughter in the auxiliary symbol and optionally the preceding sister node.
An unlexicalized parser which combines the annotation strategies mentioned above with further annotations, like the marking of non-recursive NPs, possessive NPs, non-verbal constituents dominating a verb, and S-nodes with an empty subject is described in (Klein and Manning, 2003). It achieves an accuracy of over 86% f-score (compared with 72% for the plain PCFG approach).

Subcategorization In order to enforce subcategorization restrictions, the verbs in the treebank can be annotated with a subcat feature which encodes the set of arguments (Schulte im Walde et al., 2001). An example is the rule VP → V/NP-PP NP PP which makes sure that verbs of the category V/NP-PP take an NP and a PP argument. Such a subcat feature can also be automatically annotated in the treebank.

Empty Nodes The Penn treebank contains empty nodes which encode traces of moved constituents as well as empty constituents which are stipulated for theoretical reasons (like a subject in imperative clauses). Traces are co-indexed with their fillers. Empty nodes are usually removed from a treebank before a parser is trained. Exceptions are (Collins, 1997) where some of the empty nodes are used and (Dienes and Dubey, 2003) who extend Collins’ approach. The linking of traces and fillers is done with gap threading. Adding information about empty nodes to the grammar not always improves the accuracy of the parser, but it provides valuable information for semantics construction.

Feature Induction While the annotation strategies described so far rely on features which represent well-defined information like pre-terminal heads, parent categories etc., there have also been attempts to induce features automatically (see e.g. Prescher, 2005; Koo and Collins, 2005; Matsuzaki et al., 2005). These methods are less treebank-specific and less dependent on linguistic expertise.

Lexicalization An important extension of the basic PCFG parsing approach is lexicalization. Lexicalized PCFGs (see e.g. Carroll and Rooth, 1998; Charniak, 1997; Collins, 1997) assign a lexical head to each constituent. The probability of a grammar rule depends on the lexical head of the expanded constituent which accounts for the fact that a VP, for example, is more likely to be expanded by the rule VP → V NP if the lexical head is buy than if it is sleeps. The lexicalized rule probabilities therefore encode information about subcategorization. Lexicalized PCFGs comprise another type of parameters: The lexical choice probability \( p(\text{book}|NP, VP, \text{read}) \) represents the probability that the lexical head of an NP is book if the NP is dominated by a VP with lexical head read. These parameters encode selectional restrictions. The lexicalization of PCFGs drastically increases the number of parameters that have to be estimated from training data. Therefore it is essential to smooth the parameters in order to avoid sparse data problems.

Reranking Strategies Recently, a two-step approach to statistical parsing has also become popular (Collins, 2000; Charniak and Johnson, 2005; Collins and Koo, 2005). In the first step, a lexicalized PCFG parser is used to extract the N best parse trees (where N is e.g. 100). The second step scores the different analyses (often with a log-linear probability model) and selects the final result. The advantage of the two-step approach is that the
The disambiguation algorithm of the second step is free to use arbitrary features of the parse trees (like the feature “Has this constituent a strictly right-branching structure?”). This is not possible in the first step because non-local features cannot be used with the dynamic programming methods that have to be applied there in order to deal with the huge number of possible analyses. A similar strategy is used in LFG parsing (cf. project D2) where the set of possible analyses is first filtered with OT constraints, and the D5 Project of this SFB will follow this approach, too.

**German Treebank Parsing** Treebank parsing techniques were also applied to the German Negra treebank (Dubey and Keller, 2003; Fissaha et al., 2003; Schiehlen, 2004), but the parsing accuracies are lower than for English. Dubey and Keller (2003) report that the accuracy of the Collins (1997) parser, one of the best lexicalized treebank parsers for English at that time, was actually below that of a simple unlexicalized parser when trained on the Negra treebank. This result indicates that parsing strategies developed for one treebank cannot be simply transferred to other treebanks.

### 3.3.2 Eigene Vorarbeiten

Statistical parsing techniques were extensively studied at IMS in the past (Beil et al., 1999; Schmid and Schulte im Walde, 2000; Schmid and Rooth, 2001; Schmid, 2004). Unlexicalized and Head-Lexicalized PCFG parsers have been implemented (Schmid, 2000) and used to extract lexical information from corpora using manually developed grammars (Carroll and Rooth, 1998; Schulte im Walde et al., 2001). The lexical frequency information extracted with the parsers was also used to induce verb classes with hard clustering methods (Schulte im Walde and Brew, 2002) as well as soft clustering methods (Rooth et al., 1999).

The IMS also worked on treebank parsing. Schiehlen (2004) developed a treebank parser for the Negra treebank with parent encoding, markovization, gap threading and other automatically annotated features. Treebank parsers have to deal with a huge degree of ambiguity. Therefore it is difficult to compute the whole set of possible analyses rather than just the most probable analysis, as most of the treebank parsers do. Schmid (2004) describes BitPar, a bit-vector implementation of a PCFG parser for treebank grammars which computes a compact parse forest representation of the set of analyses. Parse forests are needed by methods like the Inside-Outside algorithm which is used for unsupervised training, extraction of word-governor relations (Schmid and Rooth, 2001) and other applications. BitPar was used e.g. in (Schiehlen, 2004) and (Prescher, 2005).

Recently, the IMS has also developed an unlexicalized treebank parser\(^3\) for the Penn treebank. The PCFG was extracted from a version of the Penn treebank which was automatically annotated with additional features. The IMS approach resembles that of Klein and Manning (2003), but adds more information including subcategorization features, retains empty nodes and performs gap threading to link traces to fillers. The parsing is done with BitPar. The parsing accuracy (87.32% f-score) is slightly better than that of Klein and Manning (2003). Figure 3.1 shows an example of an annotated parse tree\(^4\).

---

\(^3\)This work by the Projektleiter has not been published, yet.

\(^4\)The feature \textit{fin} marks finite VP- and S-nodes. Non-verbal constituents dominating a verb are tagged with \textit{domV}. S-nodes dominating a subject trace are marked with a \textit{gap} feature, and verbal POS tags like \textit{VV} receive a subcat feature consisting of a sequence of letters which encode the syntactic categories of
Lexicalized PCFGs usually comprise a huge number of parameters. Because of sparse data problems, only a tiny fraction of them can be estimated reliably. Therefore parameter smoothing techniques play a key role in most lexicalized parsers. The lexicalization strategy presented in (Schmid, 2002) avoids the sparse data problem and reduces the number of parameters that have to be stored. Unlexicalized PCFGs assume that the selection of a word depends only on the part of speech $C$. The model proposed in (Schmid, 2002) assumes instead that a word $w$ also depends on its lexical governor $g$. The weakening of the independence assumption (which means that the probability $p(w|C)$ is replaced by $p(w|g, C)$) introduces lexical factors into the probability model, as the following equation shows:

$$p(w|g, C) = \underbrace{p(w|C)}_{\text{unlexicalized prob.}} \frac{p(w, g|C)}{p(w|C) p(g|C)} = L(w, g|C)$$

The unlexicalized model and the lexicalized model differ by the lexical association scores $L(w, g|C)$ which have to be multiplied for each word. These scores are defined as the ratio of the joint probability of $w$ and $g$ given $C$ divided by the product of the marginal probabilities of $w$ and $g$ given $C$. The measure is closely related to the pointwise mutual information. The lexicalized probability of the parse tree $t$ is defined as follows:

$$p(t) = p_{unlex}(t) \prod_{i=1}^{n} L(w_i, g_i|C_i)$$

$$L(w, g|C) = \frac{p(w, g|C)}{p(w|C) p(g|C)}$$

$p_{unlex}(t)$ is the unlexicalized probability of parse tree $t$, defined in the usual way as the product of the probabilities of the rules applied in the derivation of the parse tree. The unlexicalized probability of the parse tree $(S (NP he) (VP (V read) (NP (DT a) (N book))))$, e.g. is the product of the probabilities of the rules $S \rightarrow NP VP$, $NP \rightarrow he$, the arguments. Fillers are marked with $+XP$, and traces and their ancestor nodes are marked with $\_XP$.  

$^5$The lexical governor of a word $w$ is the lexical head of the first ancestor node of $w$ which is not headed by $w$.  

Abbildung 3.1: Sample parse of the English treebank parser
VP → V NP, and so on, and the lexical factors for this parse tree are $L(\text{he, read}|NP)$, $L(\text{book, read}|NP)$, $L(\text{the, book}|\text{DT})$, etc. The model is flexible wrt. the definition of the lexical governor. If the governor is defined as a pair consisting of the category and the lexical head of the first ancestor node which is not headed by the current word, the model is able to distinguish between subjects (where the ancestor is an S node) and objects (where the ancestor is a VP).

The proposed lexicalization method is strictly modular in the sense that a standard unlexicalized probabilistic context-free grammar (PCFG) is extended towards a lexicalized model by multiplying additional lexical factors. The lexical factors are only estimated for the subset of word-governor pairs whose observed frequency differs significantly from the expected value under the assumption of independence. If the word $w$ and its governor $g$ are statistically independent given the part of speech $C$, then $p(w|C)p(g|C) = p(w, g|C)$ holds by definition and therefore $L(w, g|C) = 1$. Therefore, the model assigns a default lexical factor close to 1 to all these insignificant word-governor pairs. The actual value of the default score has to be chosen in such a way that a proper probability distribution over parse trees is obtained.

Schmid (2002) proposed another lexicalization strategy in which each word depends on the lexical heads of the arguments rather than the governor, allowing the parameters to model correlations between arguments. A person e.g. is likely to cut a slice of bread, but a company is more likely to cut costs. A drawback of the second method is that the sparse data problem increases because word triples and even quadrupels have to be considered. To counter the sparse data problem, the following submodel was proposed for the probabilities of a predicate $w$ with argument frame $f$ and arguments $a_1, ..., a_{nf}$, from which the lexical factors are derived:

$$p(w, f, a_1, ..., a_{nf}) = \sum_{c \in C} p(c) \prod_{i=1}^{nf} \sum_{r \in H} p(r|c, f, i) p(a_i|r)$$

The model is an extension of the clustering model of Rooth et al. (1999). It assumes that each word $w$ belongs to one or more word clusters $c$ which correspond to different readings of the word, and that the syntactic properties and selectional properties of a particular word reading are fully described by the respective cluster probabilities (namely the frame probabilities $p(f|c)$ and the argument slot probabilities $p(r|c, f, i)$). Furthermore, the model assumes that the different arguments are statistically independent given the cluster $c$ and that the selectional properties of an argument slot are adequately described by a small set of more or less general concepts $r$ from a concept hierarchy $H$ (e.g. from WordNet). This model tackles the sparse data problem (1) by generalizing words to clusters of words with similar argument selection properties, (2) by generalizing arguments to concepts, and (3) by assuming independence between the arguments given the word reading.
3.4 Planung des Teilprojekts (Ziele, Methoden, Arbeitsprogramm)

3.4.1 Fragestellung

The main question to be answered by this project is "How can we improve upon the state of the art in the specification of syntactic structures with statistical methods using contextual information extracted from treebanks, raw texts and taxonomies?" More specifically, we will investigate the following questions:

- Which linguistically motivated features of constituents in parse trees are relevant for the statistical modeling and disambiguation of language?
- How well do the proposed lexicalization methods work? Can they close the gap between the accuracy of our current unlexicalized treebank parser and the accuracy of state-of-the-art lexicalized parsers?
- To what extent does unsupervised training on unlabeled data improve the performance of a lexicalized parser?
- To what extent do generic lexical resources like WordNet help in statistical parsing?
- To what extent is it possible to induce lexical information about verb readings, subcategorization frames, selectional preferences (and perhaps also about alternations and light verb constructions) from raw text?

3.4.2 Ziele

The main goal of this project is to advance the state of the art in the specification of syntactic structures using statistical methods. This goal is addressed by (i) improving the statistical modeling of language with better symbolic grammars which are derived from treebanks with additional automatically annotated features and by (ii) improving the statistical models with two novel lexicalization strategies.

- Development of a program for the annotation of a German treebank with additional linguistically motivated features
- Development of a lexicalized treebank parser for English, German and optionally other languages
- Implementation of an unsupervised training algorithm for the lexical parameters of the parser
- Development of a statistical soft-clustering model which models word-argument dependencies and induces verb readings, subcategorization frames and selectional restrictions
- Evaluations of the systems
3.4.3 Methoden und Arbeitsprogramm

The statistical model of the lexicalized treebank parser that will be developed in this project consists of two components. The first component is a standard unlexicalized probabilistic context-free grammar which is extracted from a treebank. The treebank is automatically annotated with additional features in order to provide important information for disambiguation (see also the section on Stand der Forschung). Some of these features might also be used in project B3 if they turn out to be useful for their parser, as well. The second component of the statistical model is a set of lexical parameters encoding dependencies between lexical heads of constituents. We will explore two types of lexical parameters, bi-lexical word-governor dependencies and word-argument set dependencies.

Word-Governor Dependencies (Model 1)

The word-governor parameters $L(w, g|C)$ have been discussed quite extensively in section Vorarbeiten. These parameters are estimated for all word-governor pairs whose observed frequency is significantly higher than their expected frequency under the assumption that $w$ and $g$ are statistically independent. A statistical test will be applied to select the set $W_{g,C}$ of significant words of category $C$ for each governor $g$. The size of this set determines the number of parameters and is influenced by the significance level. Requiring a high level of significance reduces the number of parameters. The optimal level should be determined with an evaluation on held-out data.

The lexical scores of the significant word-governor pairs are estimated according to the following formula where $f(\alpha)$ represents the frequency of the tuple $\alpha$:

$$L(w, g|C) = \frac{p(w, g|C)}{p(w|C) p(g|C)} = \frac{f(w, g, C) f(C)}{f(w, C) f(g, C)}$$

The default scores $L(w, g|C) = L(\ast, g|C)$ assigned to insignificant words $w$ are defined such that $p(w|g, C)$ is a proper probability distribution, i.e. the following equations must hold: $\sum_w p(w|g, C) = \sum_w p(w|C) L(w, g|C) = 1$. The formula for the computation of $L(\ast, g|C)$ is derived as follows:

$$\sum_w p(w|C) L(w, g|C) = 1$$

$$\sum_{w \in W_{g,C}} p(w|C) L(w, g|C) + \sum_{w \notin W_{g,C}} p(w|C) L(\ast, g|C) = 1$$

$$L(\ast, g|C) = \frac{1 - \sum_{w \in W_{g,C}} p(w|C) L(w, g|C)}{1 - \sum_{w \in W_{g,C}} p(w|C)}$$

It might be useful to also add lexical parameters for word-governor pairs whose frequency is significantly below the expected value. Such a negative correlation is much harder to detect, however. These bilexical parameters will also be used in project D2 where they will be added to the set of features of the log-linear model.
Word-Argument Set Dependencies (Model 2)

The second type of lexical parameters models the dependencies between a word $w$ of category $C$ and its argument set $A$. This model drops the assumption that the arguments are mutually independent. The lexical parameters are here defined as follows:

$$L(w, A | C) = \frac{p(w, A | C)}{p(w | C) p(A | C)}$$

The probabilities $p(w, A | C)$, $p(w | C)$, and $p(A | C)$ are derived from a statistical sub-model for word-argument tuples. This sub-model groups all the words into clusters of words with similar subcategorization and selectional properties. Words may be assigned to several clusters (soft clustering) which allows the model to describe the syntactic properties of several word readings separately. The number of word clusters is defined in advance, but the assignment of the words to the clusters is learned during training. It is assumed that all words (or rather word readings) belonging to one cluster have identical subcategorization and selectional properties. This assumption reduces the number of parameters in the model and increases the amount of data available for the estimation of each parameter. The selectional properties are expressed in terms of semantic concepts (from a conceptual hierarchy like WordNet) rather than a set of individual words, which also decreases the size of the parameter set. Finally, the model assumes that the different arguments are mutually independent for all subframes of a cluster, once more reducing the parameter set. The last assumption entails that any statistical dependency between the arguments of a word has to be explained with multiple readings.

The statistical sub-model is characterized by the following formula which defines the probability of a word $w$ with subcat frame $f$ and arguments $a_1, ..., a_{nf}$:

$$p(w, A) = p(w, f, a_1, ..., a_{nf}) = \sum_c p(c) p(w | c) p(f | c) \prod_{i=1}^{nf} \sum_{r \in R} p(r | c, f, i) p(a_i | r)$$

The model describes a stochastic process which generates a word argument tuple like \langle speak, NP-PPto, professor, audience \rangle by

1. selecting some cluster, e.g. $C_3$ (which might correspond to a set of communication verbs), with probability $p(C_3)$
2. selecting a word, here the word `speak`, from cluster $C_3$ with probability $p(\text{speak} | C_3)$
3. selecting a subcat frame, here NP-PPto, with probability $p(\text{NP-PPto} | C_3)$, (Note that the frame probability only depends on the cluster, not on the word.)
4. selecting a WordNet concept for each argument slot, e.g. Person for the first slot with probability $p(\text{Person} | C_3, \text{NP-PPto}, 1)$ and SocialGroup for the second slot with probability $p(\text{SocialGroup} | C_3, \text{NP-PPto}, 2)$
5. choosing a word to instantiate each concept. In our example, we might choose professor for Person with probability $p(\text{professor} | \text{Person})$ and audience for SocialGroup with probability $p(\text{audience} | \text{SocialGroup})$
The model contains two hidden variables, namely the clusters $c$ and the selectional restrictions $r$. In order to obtain the overall probability of a given word-argument tuple, we have to sum over all possible values of these hidden variables.

Our model is an extension of the latent semantic clustering (LSC) model (Rooth et al., 1999) for verb-object pairs which is characterized by the formula:

$$ p(v, o) = \sum_c p(c)p(v|c)p(o|c) $$

The LSC model only considers a single argument (or a fixed number of arguments from one particular subcat frame), whereas our model defines a probability distribution over all subcat frames. Furthermore it specifies selectional restrictions in terms of general WordNet concepts rather than sets of individual words.

**Training**  Model 2 will be trained on word-argument tuples which we extract from parsed corpora using either the unlexicalized treebank parser or the lexicalized treebank parser with bootstrapping, or some other parser. Because of the hidden variables, Model 2 has to be trained iteratively with the Expectation-Maximization (EM) algorithm. The parameters are randomly initialized and then re-estimated with the Inside-Outside algorithm (Lari and Young, 1990) which is an instance of the EM algorithm for PCFG training.

The PCFG training algorithm is applicable here because we can define a PCFG for each of our models which generates the same word-argument tuples with the same probability. The PCFG is defined as follows:

- The start symbol is TOP.
- For each cluster $c$, we add a rule $\text{TOP} \rightarrow V_c A_c$ whose probability is $p(c)$.
- For each word $w$ in cluster $c$, we add a rule $V_c \rightarrow w$ with probability $p(w|c)$
- For each subcat frame $f$ of cluster $c$ with length $n_f$, we add a rule $A_c \rightarrow f R_{c_{f1}} \ldots R_{c_{fn_f}}$ with probability $p(f|c)$
- For each selectional restriction $r$ of slot $i$ of subcat frame $f$ of cluster $c$, we add a rule $R_{c_{fi}} \rightarrow r$ with probability $p(r|c, f, i)$
- For each argument head $a$ which is an instantiation of concept $r$, we add a rule $r \rightarrow a$ with probability $p(a|r)$
- States with no outgoing transitions are terminal states.

A “parse” for $\langle\text{speak NP-PP} \text{to professor audience} \rangle$ is shown in figure 3.2.

The EM training algorithm maximizes the likelihood of the training data. A model with a large number of fine-grained concepts as selectional restrictions is able to assign a higher likelihood to the data than a model with a small number of general concepts. Therefore, the EM algorithm will always choose the most fine-grained concepts and – due to sparse data problems – it will overfit the training data.
In order to find selectional restrictions with a more appropriate granularity, we will apply the minimum description length (MDL) principle. The MDL principle says that the model with minimal description length should be chosen. The description length is the sum of the model length and the data length. The model length is the number of bits required to encode the model and its parameters. The data length is the number of bits required to encode the training data with the given probability model. According to coding theory, an optimal encoding uses \(-\log_2 p\) bits, on average, to encode data whose probability is \(p\). Usually, the model length increases and the data length decreases as more parameters are added to a model. The MDL principle finds a compromise between the size of the model and the accuracy of the data description.

Following Abney and Light (1999), we map WordNet to a Markov model. Each word or synset \(x\) corresponds to a state \(s_x\) of the Markov model. The transition probability from state \(s_x\) to state \(s_y\) is non-zero, iff either \(x\) is a (direct) hypernym of \(y\) or if \(y\) is a word in synset \(x\). The start state of the Markov model is an extra state with non-zero transition probabilities to all states \(s_x\) where \(x\) is a top-level synset without hypernyms. The probability of a path in the WordNet Markov model is the product of the probabilities of the transitions on the path.
Given a WordNet Markov model, we can define the probability of a concept $r$ as the sum of the probabilities of all paths starting at the start state and ending at state $s_r$. Similarly, the probability of a lexical head $a$ given some concept $r$ is the sum of the probabilities of all paths starting at state $s_r$ and ending at state $s_a$.

Model 2 comprises one (complete) WordNet Markov model for the lexical head probabilities $p(a|r)$ and one (partial) model for each selectional probability distribution $p(r|c, f, i)$. The Markov models for the selectional probabilities only contain the states which correspond to the concepts acting as selectional restrictions and their (direct and indirect) hypernyms. Figure 3.4 shows an example.

![Diagram](top)

Abbildung 3.4: A selectional restriction model

**MDL principle and EM training**  The MDL principle is applied to prune the selectional restriction models. It is integrated into the EM training which consists of the following steps:

1. Estimation of the expected frequencies using the Inside-Outside algorithm. The following frequencies are needed:
   - cluster frequency $F_c$
   - cluster-word frequency $F_{c,w}$
   - cluster-frame frequency $F_{c,f}$
   - frequency $F_{c,f,i,x}$ of visiting Markov state $x$ when generating an argument for slot $\langle c, f, i \rangle$
   - frequency $F_{c,f,i,x,y}$ of the transition from Markov state $x$ to state $y$ when generating an argument for slot $\langle c, f, i \rangle$
   - overall frequency $F_x$ of visiting Markov state $x$
   - overall frequency $F_{x,y}$ of the transition from Markov state $x$ to state $y$

2. Estimation of the probabilities with relative frequencies

$$p(c) = \frac{F_c}{\sum_{c'} F_{c'}}$$
$$p(w|c) = \frac{F_{c,w}}{F_c}$$
$$p(f|c) = \frac{F_{c,f}}{F_c}$$
\[
p_{c,f,i}(r, r') = \frac{F_{c,f,i,x,y}}{F_{c,f,i,x}}
\]
\[
p(r, r') = \frac{F_{x,y}}{F_x}
\]

3. Pruning of the selectional restriction models with the MDL principle

**MDL Pruning** The second step of the EM training returns selectional restriction models which are maximally specific. The MDL training generalizes these models by pruning the graph. The pruning algorithm examines each pre-terminal state of the graph and turns it into a terminal state if that decreases the description length.\(^6\)

The decrease of the model length depends on the encoding of the model. We assume the following encoding: Each state has 1 bit to indicate whether it is a final state. If some state has \(n\) outgoing transitions of which \(m\) transitions are non-zero, then we store the \(m\) indices of the non-zero transitions and the \(m-1\) smallest transition probabilities. (The last probability is obtained by subtracting the other probabilities from 1. We store the smallest probabilities to minimize rounding errors.) Finally, we need to encode \(m\), the number of non-zero transitions. We do so by adding another bit for each transition which indicates whether this transition is the last one. If we use \([\log_2 n]\) bits to encode a transition index and \([0.5 \log_2 F_{c,f,i,x}]\) bits to encode a probability (see Abe and Li, 1996), then we need \(m \times (1 + [\log_2 n]) + (m - 1) [0.5 \log_2 F_{c,f,i,x}]\) bits to encode the transitions and another bit for each of the \(m\) target states to encode whether it is a final state or not.

The pruning of the graph at some pre-terminal state \(s_x\) changes the description length as follows: If \(n\) is the number of hyponyms of \(x\) and \(0 < m \leq n\) is the number of transitions from state \(s_x\), then pruning at state \(s_x\) decreases the model length by \(m \times (2 + [\log_2 n]) + (m - 1) [0.5 \log_2 F_{c,f,i,x}]\) bits. The increase in the data description length is given by \(\sum_y F_{c,f,i,x,y} (\log_2 p_{c,f,i}(y|x) - \log_2 p(y|x))\) because we replace the slot-specific transition probability \(p_{c,f,i}(y|x)\) of the selectional restriction model with the less accurate general transition probability \(p(y|x)\) of the lexical head model.

![Abbildung 3.5: Sample hierarchy](image)

We will show with a German example how the MDL training works. Consider the conceptual hierarchy of figure 3.5. For simplicity, we assume that all transitions in the Markov model for the lexical head probabilities \(p(a|r)\) have a probability of 0.5. We further assume that the selectional restriction models are fully pruned at the beginning i.e. they are empty. If the words *Bank* and *Stuhl* both appeared 10 times in the argument slot

\(^6\)Statistical tests could also be applied instead of MDL as a pruning criterion.
slot whose selectional restrictions we are learning, then we obtain the following frequency estimates:

\[
\begin{align*}
F_{\text{Entität,Objekt}} &= 15 \\
F_{\text{Entität,Institution}} &= 5 \\
F_{\text{Objekt,Stuhl}} &= 10 \\
F_{\text{Objekt,Bank}} &= 5 \\
F_{\text{Institution,Bank}} &= 5 \\
F_{\text{Institution,Firma}} &= 0
\end{align*}
\]

From these frequencies, we derive the following selectional probabilities for the argument slot slot:

\[
\begin{align*}
p_{\text{slot}}(\text{Objekt}|\text{Entität}) &= 3/4 \\
p_{\text{slot}}(\text{Institution}|\text{Entität}) &= 1/4 \\
p_{\text{slot}}(\text{Stuhl}|\text{Objekt}) &= 2/3 \\
p_{\text{slot}}(\text{Bank}|\text{Objekt}) &= 1/3 \\
p_{\text{slot}}(\text{Bank}|\text{Institution}) &= 1 \\
p_{\text{slot}}(\text{Firma}|\text{Institution}) &= 0
\end{align*}
\]

Pruning the model at the Objekt node decreases the model length by \(2 \times (1 + 1 + 1) + 1 \times \lceil 1/2 \log_2 15 \rceil \) = 8 bits. The length of the data encoding increases by \(10 \times (\log_2 2/3 - \log_2 1/2) + 5 \times (\log_2 1/3 - \log_2 1/2) \approx 1.23\) bits. Overall the description length decreases by 6.77 bits, so we should prune here.

Pruning the model at the Institution node decreases the model length by \(1 \times (1 + 1 + 1) + 0 = 3\) bits\(^7\). The length of the data encoding increases by \(5 \times (\log_2 1 - \log_2 1/2) = 5\) bits. Overall the description length increases by 2 bits, so we should not prune here.

\[
\begin{tikzpicture}
  \node {<Entität>} child {node {<Objekt> \nodepart{two} 3/4 \nodepart{three} 1/4} child {node {<Institution> \nodepart{two} 1}} child {node {Bank}}};
\end{tikzpicture}
\]

Abbildung 3.6: Selectional restriction model

After the MDL pruning, we obtain the selectional restriction model shown in figure 3.6. In the second iteration, we observe a shift in the estimated frequencies.

\[
\begin{align*}
F_{\text{Entität,Objekt}} &= 16 \\
F_{\text{Entität,Institution}} &= 4 \\
F_{\text{Objekt,Stuhl}} &= 10
\end{align*}
\]

\(^7\)Note that the Institution node has only 1 non-zero child node for which no probability is stored.
The *Objekt* concept becomes more frequent because it is preferred by the selectional restriction model. With each iteration, the preference of the selectional restriction model for the *Objekt* concept increases and in the end, we obtain the model displayed in figure 3.7. This example shows that the MDL training not only finds the appropriate level of generalization, but also explains away spurious ambiguities.

\[
\begin{align*}
F_{\text{Objekt, Bank}} &= 6 \\
F_{\text{Institution, Bank}} &= 4 \\
F_{\text{Institution, Firma}} &= 0
\end{align*}
\]

### Extraction of Lexical Information

We anticipate that the trained version of Model 2 will provide valuable information for lexical research. Rooth et al. (1999) showed that the soft-clustering approach with EM training works and is able to induce classes of semantically related words. We expect that our model will do even better because it includes information about all subcategorization frames and it should be less affected by sparse-data problems due to the generalization of arguments to concepts. We will extract the following types of information:

- **Verb Classes**: Levin (1993) defined English verb classes on the basis of their subcategorization properties. Model 2 also clusters words according to their subcategorization and selectional behavior. So, we expect to obtain clusters which resemble the Levin classes. A cluster \( c \) is characterized by the words \( w \) with the highest conditional probability \( p(w|c) \).

- **Word Readings**: According to the formula \( p(c|w) = \frac{p(w|c)p(c)}{p(w)} = \frac{p(w|c)p(c)}{\sum_{c'} p(c')p(w|c')} \), we obtain cluster probabilities for each word which will be used to identify word readings. If the number of non-zero cluster probabilities is large, we could ignore clusters with a probability below some threshold (e.g. below 1%).

- **Subcat Frames**: The parameters \( p(f|c) \) encode the subcat frame probabilities of the words belonging to cluster \( c \). The generalization to word clusters has a smoothing effect on the subcat frame probabilities: If the word \( w \) belonging to cluster \( c \) did not occur with some frame \( f \), but other words of the same cluster did, the joint probability \( p(w, f|c) = p(w|c)p(f|c) \) will be non-zero. In other words, the model infers missing subcat frames from the subcat frames of similar words, which is particularly useful for low-frequency words.

- **Selectional Restrictions**: The selectional restrictions of a word are derived from the selectional restrictions of the word clusters to which it belongs, that is, from the
parameters \( p(r|c, f, i) \). Again, word clustering improves the acquisition of selectional restrictions for low-frequency words.

- Augmentation of WordNet: The WordNet hierarchy is necessarily incomplete. We can deal with unknown words in the following way. We add the unknown word \( a \) tentatively to every WordNet synset \( r \), i.e. \( p(a|r) \) is non-zero for all \( r \). After the training, the unknown word will have a preference for some WordNet synsets and will be deleted from the other synsets. The remaining synsets are the readings of the unknown word.

- Collocations: Collocations and fixed expressions like *to kick the bucket* are characterized by the fact that the elements cannot be replaced by semantically similar words without changing the meaning drastically. *to push the bucket* or *to kick the pail*, for instance, do not have the meaning *to die* and they are also less frequent than *to kick the bucket*. However, the replaceability of the elements is an essential property of the word clusters in model 2. Therefore collocations are not adequately described and the frequency predicted by the model is too low. This property was exploited in Prescher and Heid (2000) to extract collocations.

In order to improve the modeling of collocations, we can identify verb-object pairs whose empirical probability is much higher than the probability assigned by model 2. Such verb-object pairs could be replaced by a complex intransitive verb. That would allow the model to assign the collocation *to kick the bucket* to the same cluster of intransitive words as *to die*.

- Alternations: Some word classes show systematic dependencies between two subcategories. The sentence *Peter gave Mary the book* e.g. can be rephrased as *Peter gave the book to Mary*. Dative shift verbs like *to give* show the same selectional restrictions in the indirect object-slot of the NP-NP frame and in the PPto-slot of the NP-PPto frame.

In order to identify dative shift verbs, we can tentatively merge the selectional restriction models of the two slots for each cluster. If the model merging reduces the description length, we can conclude that verbs belonging to this cluster are dative shift verbs. Other alternations can be dealt with in the same way.

The soft-clustering model resembles the purely symbolic verb hierarchy which will be developed in project B5. Both of them group the verbs according to their subcategorization and selectional restrictions. We would like to check in collaboration with project B5 how well the verb groups created by the soft-clustering approach match the criteria established for verb groups in project B5. Furthermore, we will pass information about the subcategorization and selectional restrictions of verbs to projects B3 and B4.

Project D5 will use our English treebank parser\(^8\) as the baseline parser. They will develop disambiguation methods which improve upon the baseline parser by reranking its best analyses using information which the statistical model of our parser fails to represent.

---

\(^8\)At first, they will use the unlexicalized parser, but they will switch to the lexicalized version once a stable version of it becomes available.


