12

Statistical Grammar Models and Lexicon Acquisition

S. Schulte i. W., H. Schmid, M. Rooth, S. Riezler, D. Prescher

12.1 Introduction

This paper presents a framework for developing and training statistical grammar models for the acquisition of lexicon information. Utilising a robust parsing environment and mathematically well-defined unsupervised training methods, the framework enables us to induce lexicon information from text corpora. Particular strengths of the approach concern (i) the fact that no extensive manual work is required to set up the framework, and (ii) that the framework is applicable to any desired language. It has already been applied to English and German (Carroll and Rooth 1998, Beil et al. 1999, Rooth et al. 1999, Schulte im Walde 2000a), Portuguese (de Lima 2001), and Chinese (Hockenmaier 1999).

Manual work within the framework is reduced to a minimum, since the necessary grammars need not go into detailed structures for the relevant grammar aspects to be trained sufficiently. The automatic training process utilises a shallow parser embedded in the mathematically well-defined Expectation-Maximisation algorithm. The training approach enforces the lexicalised parameters in the statistical grammar to obtain linguistic reliability. A basic assumption thereby expects that the linguistically correct analyses of text correspond to those analyses which

Linguistic Form and its Computation.
Edited by
Christian Rohrer, Antje Roßdeutscher and
Hans Kamp.

maximise the probability of the data.

The linguistic value of the grammar models mainly lies in the lexicalised model parameters: they contain lexicalised rules, i.e. grammar rules referring to a specific lexical head, and lexical choice parameters, a measure of lexical coherence between lexical heads. Concerning verbs, for example, the lexical rule parameters serve as basis for probability distributions over subcategorisation frames, and the lexical choice parameters supply us with nominal heads of subcategorised noun phrases, as basis for selectional constraints. The information can be used straightly as lexical description, or as input for lexicon tools, such as semantic clustering techniques (Rooth et al. 1999, Schulte im Walde 2000a), or as basis for a variety of applications, e.g. parser improvement (Riezler et al. 2000), chunking (Schmid and Schulte im Walde 2000), or machine translation (Prescher et al. 2000).

The reader might still wonder about the exact nature of the lexical information we gain. Consider this concrete example: our trained grammar model for German informs you that the verb *essen* 'eat' most probably occurs transitively, but might as well occur intransitively. In addition, we learn that e.g. the most frequent nominal heads in the direct object slot of the transitive frame are the German equivalent nouns for *bread*, *meat*, *banana* and *ice-cream*.

The first part of this chapter concerns the grammar development and its training: section 12.2 allows practical insights into the prerequisites for our statistical grammars and describes a characteristic grammar development process by means of the German grammar. Following in section 12.3, the reader will find an introduction to the theoretical background of statistical grammars and their head-lexicalised refinements, as well as a description of their training facilities. Section 12.4 then presents the application of the training procedure concerning the German grammar example.

The second part of this chapter illustrates various possibilities to exploit the lexicalised probability models: section 12.5 straightly utilises the model parameters, to extract lexical parameters for –mainly– verbs, and to apply specific parsing facilities such as Viterbi parsing, or noun chunking. Section 12.6 demonstrates the usage of lexical information –with specific reference to lexical coherence between verbs and subcategorised nouns– as input for semantic clustering techniques.

12.2 Grammar Development

Our statistical grammar models can be developed for arbitrary languages, presupposing (i) a corpus as source for empirical input data,

(ii) a morphological analyser for analysing the corpus word-forms and assigning lemmas where appropriate, and (iii) a context-free grammar (CFG) for parsing the corpus data.

The grammar is supposed to cover a sufficient part of the corpus, since in order to develop a statistical grammar model on basis of the grammar (cf. sections 12.3 and 12.4), a large amount of structural relations within parses is required. The more corpus data is accessible for grammar training, the more reliable the probability model will be.

As mentioned in the introduction, manual work concerning the grammar is reduced to a minimum. The necessary grammars need not go into detailed structures for the relevant grammar aspects to be trained sufficiently. The complete framework can be set up within a few weeks time, and easily be transferred to a different language. This property advances the grammar framework compared to e.g. tree-bank grammars (Charniak 1996), since it does not presuppose a tree-bank for the relevant language.

So far, we have worked on statistical grammar models for English (Carroll and Rooth 1998), German (an earlier version is described in (Beil et al. 1999)), Portuguese (de Lima 2001), and Chinese (Hockenmaier 1999). The preparation of the relevant corpus data, the task definition of the morphological analyser and a context-free grammar are described below. For the purpose of illustrating the grammar development framework, we concentrate on the German model. We specifically describe the grammar development facilities and outline the grammar structure.

12.2.1 Corpus Preparation

We created two sub-corpora from the 200 million token newspaper corpus *Huge German Corpus (HGC)*, (a) a sub-corpus containing 450,000 verb-final clauses with a total of 4 million words, and (b) a sub-corpus containing 1,1 million relative clauses with a total of 10 million words. Apart from non-finite clauses as verbal arguments, there are no further clausal embeddings, and the clauses do not contain any punctuation except for a terminal period. The average clause length is 9.16 and 9.12 words per clause, respectively.

12.2.2 Morphological Analyser

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

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

Reducing the ambiguous categories leaves the two morphological analyses

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

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

12.2.3 The German Context-Free Grammar

The context-free grammar contains 5,033 rules with their heads marked. With very few exceptions (rules for coordination, S-rule), the rules do not have more than two daughters. The 220 terminal categories in the grammar correspond to the collapsed corpus tags assigned by the morphology.

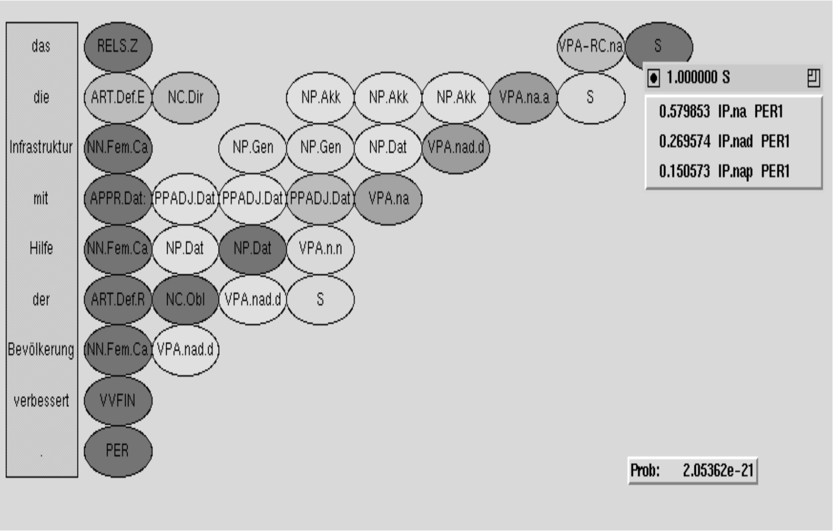
Grammar development is facilitated by (a) grammar development environment of the feature-based grammar formalism YAP (Schmid 1999), and (b) a chart browser that permits a quick and efficient discovery of grammar bugs (Carroll 1997). Figure 1 shows that the ambiguity in the chart is quite considerable even though grammar and corpus are restricted.

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

The following sections describe two essential parts of the grammar, the noun chunks and the definition of subcategorisation frames. For more details concerning the German grammar structure, see (Schulte im Walde 2000b).

Noun Chunks

On nominal categories, in addition to the four cases `Nom`, `Gen`, `Dat`, and `Akk`, case features with a disjunctive interpretation (such as `Dir` for `Nom` or `Akk`) are used. The grammar is written in such a way that non-disjunctive features are introduced high up in the tree. Figures 2 to 5 illustrate the use of disjunctive features in the noun projections for the



```
● 1.000000 S
0.579853 IP.na PER1
0.269574 IP.nad PER1
0.150573 IP.nap PER1
```

```
Prob: 2.05362e-21
```

FIGURE 1 Chart Browser for Grammar Development

German noun phrase *eine gute Gelegenheit* 'a good opportunity' in all four cases; the terminal `NN` contains the four-way ambiguous `Cas` case feature; the `N`-bar (`NN1`) and noun chunk `NC` projections disambiguate to two-way ambiguous case features `Dir` and `Obl`; the weak/strong (`Sw/St`) feature of `NN1` allows or prevents combination with a determiner, respectively; only at the noun phrase `NP` projection level, the case feature appears in disambiguated form. The use of disjunctive case features results in some reduction in the size of the parse forest. Essentially the full range of agreement inside the noun phrase is enforced. Agreement between the subject `NP` and the tensed verb is not enforced by the grammar, in order to control the number of parameters and rules.

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

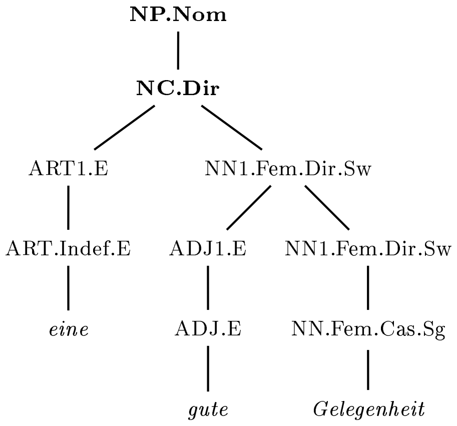


FIGURE 2 Noun Projection: NP with Nominative Case

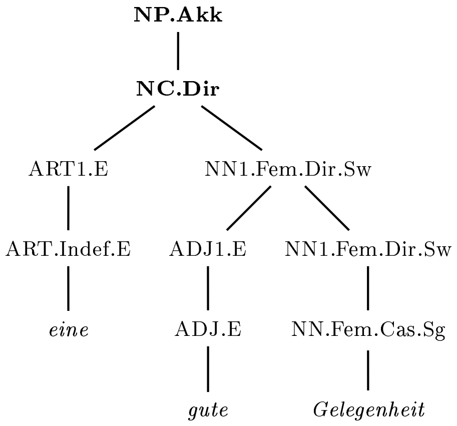


FIGURE 3 Noun Projection: NP with Accusative Case

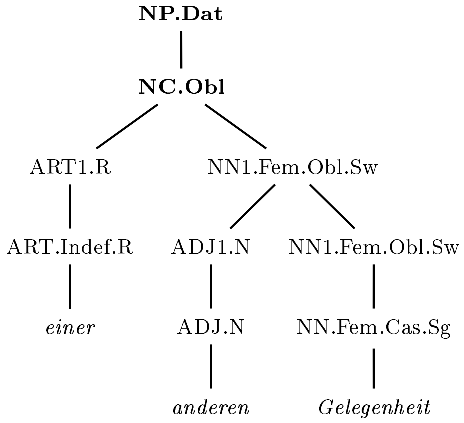


FIGURE 4 Noun Projection: NP with Dative Case

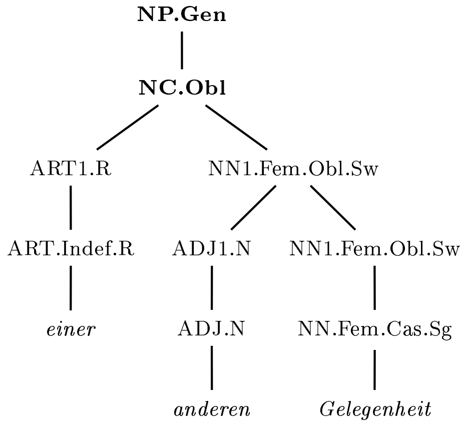


FIGURE 5 Noun Projection: NP with Genitive Case

Subcategorisation Frames

The grammar distinguishes four subcategorisation frame classes: active (VPA), passive (VPP), non-finite (VPI) frames, and copula constructions (VPK). A frame may have maximally three arguments. Possible arguments in the frames are nominative (n), dative (d) and accusative (a) NPs, reflexive pronouns (r), PPs (p), and non-finite VPs (i). The grammar does not distinguish plain non-finite VPs from *zu*-non-finite VPs. The grammar is designed to distinguish between PPs representing a verbal complement or adjunct: only complements are referred to by the frame type. The number and the types of frames in the different frame classes are given in Table 1.

Frame Class	#	Frame Types
VPA	16	n, na, nd, np, nad, nap, ndp ni, di, nai, ndi nr, nar, ndr, npr, nir
VPP	18	n, np-s, d, dp-s, p, pp-s nd, ndp-s, np, npp-s, dp, dpp-s i, ip-s, ni, nip-s, di, dip-s
VPI	8	-, a, d, p, r, ad, ap, dp, pr
VPK	2	n, i

TABLE 1 Subcategorisation Frame Types

German, being a language with comparatively free phrase order, allows for scrambling of arguments. Scrambling is reflected in the particular sequence in which the arguments of the verb frame are saturated. Compare Figure 6 as example of a canonical subject-object order within an active transitive frame *der sie liebt* 'who loves her' and its scrambled object-subject order *den sie liebt* 'whom she loves'.

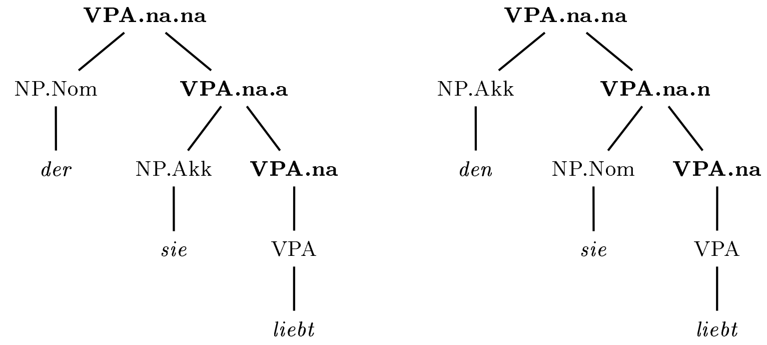


FIGURE 6 Realising Scrambling Effect in the Grammar Rules

Abstracting from the active and passive realisation of an identical underlying deep-level syntax we generalise over the alternation by defining a top-level subcategorisation frame type, e.g. `IP.nad` for `VPA.nad`, `VPP.nd` and `VPP.ndp-s` (with `p-s` a prepositional phrase within passive frame types representing the deep-structure subject, realisable only by PPs headed by *von* or *durch* 'by'); see Figure 7 as example, presenting the relative clauses *der die Frau verfolgt* 'who follows the woman', *die verfolgt wird* 'who is followed' and *die von dem Mann verfolgt wird* 'who is followed by the man'.

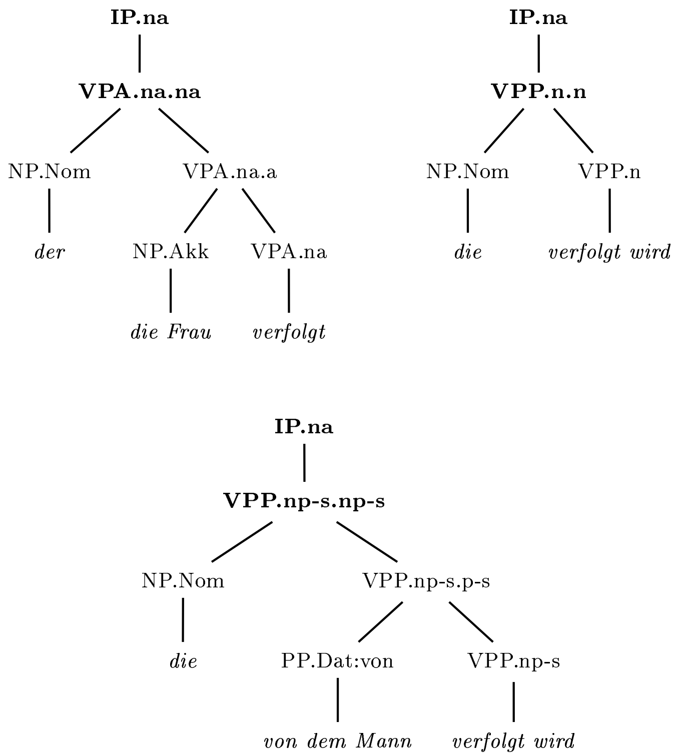


FIGURE 7 Generalising over the Active-Passive Alternation of Subcategorisation Frames

12.3 Probability Model

The probabilistic grammars are parsed with a head-lexicalised probabilistic context-free parser called `LoPar`[10](Schmid 2000). It is an implementation of the Left-Corner algorithm for parsing and of the Inside-Outside algorithm for parameter estimation. Probabilistic context-free parsing is a well-known technique (Lari and Young 1990). Innovative features of LoPar are head lexicalisation, lemmatisation, parameter pooling, and a sophisticated smoothing technique.

12.3.1 Probabilistic Context-Free Grammars

A probabilistic context-free grammar (PCFG) is a context-free grammar which additionally assigns a probability $P(r)$ to each grammar rule `r`. The probability of a parse tree is defined as the product of the probabilities of the rules which are used to build the parse tree.

PCFGs rank the different analyses (= parse trees) of a sentence according to their probabilities. However, PCFGs fail to resolve some frequent syntactic ambiguities like PP attachment ambiguities and coordination ambiguities. For example, in the sentence *The COLING conference in August at the University of Saarland in Saarbrücken was well attended*, the prepositional phrase *in Saarbrücken* could syntactically attach to any of the preceding noun phrases. Disambiguation of these ambiguities requires information about the lexical heads of the constituents (see also (Hindle and Rooth 1993)). Head-lexicalised probabilistic context-free grammars incorporate this type of information.

Head-Lexicalised Probabilistic Context-Free Grammars

Syntactically, a head-lexicalised probabilistic context-free grammar (HPCFG) (Carroll 1995, Carroll and Rooth 1998) is a probabilistic context-free grammar in which one of the categories on the right hand side of each grammar rule is marked as the head by an apostrophe (`'`), e.g. `NP` $\rightarrow$ `DT N'`. Each constituent bears a lexical head, which is propagated from the head daughter. The lexical head of a terminal node is the respective word form.

[10] LoPar is basically a re-implementation of the Galacsy tools which were developed by Glenn Carroll in the SFB, but LoPar provides additional functionality.

HPCFGs assign the following probability[11] to a parse tree T:

$$\begin{aligned} P(T) = {} & P_{start}(cat(root(T))) \; * \\ & P_{start}(head(root(T)) \mid cat(root(T))) \; * \\ & \prod_{nonterm\ n\ in\ T} P_{rule}(rule(n) \mid cat(n), head(n)) \; * \\ & \prod_{nonroot\ n\ in\ T} P_{choice}(head(n) \mid cat(n), cat(par(n)), head(par(n))) \; * \\ & \prod_{term\ n\ in\ T} P_{rule}(\langle t \rangle \mid cat(n), head(n)) \; P_{lex}(word(n) \mid cat(n), head(n)) \end{aligned}$$

Five families of probability distributions are relevant here. $P_{start}(C)$ is the probability that C is the category of the root node of a parse tree. $P_{start}(h|C)$ is the probability that a root node of category C bears the lexical head h. $P_{rule}(r|C,h)$ is the probability that a node of category C with lexical head h is expanded by rule r. $P_{choice}(h|C,C_p,h_p)$ is the probability that a (non-head) node of category C has the lexical head h given that the parent category is C_p and the parent head is h_p. $P_{rule}(\langle t \rangle|C,h)$ is the probability that a node of category C with lexical head h is a terminal node. $P_{lex}(w|C,h)$, finally, is the probability that a terminal node with category C and lexical head h expands to the word form w. If the lexical head of a terminal node is the word form itself (rather than e.g. its lemma), then $P_{lex}(w|C,h)$ is 1 if w and h are identical and 0 otherwise.

Lemmatisation

The major problem in training HPCFGs is the large number of parameters which have to be estimated from a limited amount of training data. The number of parameters is reduced if stems are used as lexical heads rather than inflected word forms, increasing the reliability of the parameter estimates. This is in particular true for languages with a rich morphology like German.

If the lexical heads are stems, the word form probability distribution $P_{lex}(w|C,h)$ is not trivial anymore because several word forms could have the same stem and part of speech (just assume that all numbers have the same stem). The P_{lex} parameters therefore have to be estimated from training data like other parameters.

[11] The auxiliary functions `cat`, `head`, `par(ent)`, `word` and `rule` return the syntactic category, the lexical head, the parent node, the dominated word or the expanding grammar rule of a node. `root` returns the root node of a parse tree and `<t>` is a constant.

12.3.2 Parameter Estimation

The parameters of lexicalised as well as unlexicalised probabilistic context-free grammars are iteratively estimated with the *Inside-Outside algorithm* (Lari and Young 1990), which is an instance of the *Expectation-Maximisation* (EM) algorithm (Baum 1972). Each iteration of the Inside-Outside algorithm consists of two steps, namely frequency estimation and parameter estimation.

Lexicalised probability models are estimated with a bootstrapping approach. First, an unlexicalised PCFG is trained starting with a randomly initialised model. The unlexicalised PCFG is then used to estimate initial values for the lexicalised probability model. The lexicalised model is retrained until it does not improve anymore.

Parameter Smoothing

The number of parameters of PCFGs and HPCFGs is usually so large that some of the corresponding events do not occur in the training data. Their estimated frequency is therefore 0. The same holds for the probabilities if relative frequency estimates are used. In order to avoid that all analyses with unobserved events are assigned zero probabilities, the probability distributions are *smoothed*. A variant of the absolute discounting method (Ney et al. 1994) is used for this purpose.

The basic idea of absolute discounting is to subtract a small amount (the discount) from all frequency counts and to redistribute the sum of these discounts over the events with zero frequency according to some *backoff* distribution. This is done recursively. The absolute discounting method had to be adapted in order to be applicable to the real-valued frequency counts generated by LoPar.

Parameter Pooling

It has already been discussed how lemmatisation is used to reduce the number of parameters of a HPCFG. Another way to achieve a reduction is *parameter pooling*. Parameter pooling applies to the lexical choice probabilities. It is based on the observation that the probability of the lexical head of the daughter node is usually similar for different inflectional variants of the lexical head of the mother node. Consider the following grammar rule which adjoins an adverb to a verb phrase.

```
VP_fin_past -> VP_fin_past' ADV
```

The lexical choice probability $P_{choice}(heavily|ADV, VP\_fin\_past, rain)$ is unlikely to differ much from the probability $P_{choice}(heavily|ADV, VP\_fin\_pres, rain)$ or $P_{choice}(heavily|ADV, VP\_inf, rain)$ etc. Therefore, it is possible to pool the corresponding distributions into one distribution

$P_{choice}(adv|ADV, VP\_fin\_past|VP\_fin\_pres|..., verb)$ in order to get more reliable estimates.

Similarly, it is possible to pool the daughter categories. By pooling mother and daughter categories in case of the rules

```
NBAR_nom_sg -> ADJ_nom_sg NBAR_nom_sg'
NBAR_nom_pl -> ADJ_nom_pl NBAR_nom_pl'
NBAR_gen_sg -> ADJ_gen_sg NBAR_gen_sg'
...
NBAR_acc_pl -> ADJ_acc_pl NBAR_acc_pl'
```

we obtain a single probability distribution for the adjectival modifiers of the German noun *Buch* 'book'. If the phrase *das alte Buch* 'the old book' (nominative case) is observed in the training data, the probability of the phrase *den alten Büchern* 'the old books' (dative case) will also be high.

12.4 Statistical Grammar Training

What is the linguistically optimal strategy for training a head-lexicalised probabilistic context-free grammar, i.e. estimating the model parameters in the optimal way? The EM-algorithm guarantees improving an underlying model towards a (local) maximum of the likelihood of the training corpus, but is that adequate for improving the linguistic representation within the probabilistic model? Various training strategies have been developed in the past years, with preliminary results referred to by Beil et al. (Beil et al. 1998).

Elaborating the optimal training strategy results from the interaction between the linguistic and mathematical motivation and properties of the probability model:

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

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

  where $P_M(C)$ is the *likelihood* of corpus C according to model M, and N is the size of the corpus. Intuitively, the perplexity measures the uncertainty about the next word in a corpus. For example, if the perplexity is 23, then the uncertainty is as high as it is when we have to choose from 23 alternatives of equal probability.
  The perplexity on the training and test data should decrease during training. At one point the perplexity on the test data will increase again which is referred to as *over-training.* The optimal point of time to stop the training is at the minimum of perplexity, before

the increase.

- Linguistic motivation: representation of linguistic features
  The linguistic parameters can be controlled by investigating rule and lexical choice parameters, e.g. what is the probability distribution over subcategorisation frames concerning the verb *achten* (ambiguous between 'to respect' and 'to pay attention'), and does it correlate to existing lexical information?
  In addition, the models were inspected by controlling the parsing performance on specified grammatical structures, i.e. noun chunks and verb phrases have been assigned labels which form the basis for evaluating parses.

Section 12.4.1 describes the up to the present optimal training strategy. In section 12.4.2 the resulting model is evaluated; section 12.4.3 describes the linguistic performance in more detail, i.e. strength and weaknesses of the model are investigated.

12.4.1 Training Strategy

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

1. Initialisation:
   The grammar was initialised by identical frequencies for all context-free grammar rules.
   Comparative initialisations with random frequencies had no effect on the model development.
2. Unlexicalised training:
   The training corpus was parsed once with LoPar, re-estimating the frequencies twice.
   The optimal training strategy proceeds with few parameter re-estimations. Without re-estimations or with a large number of re-estimations the model was effected to its disadvantage.
   With less unlexicalised training more changes during lexicalised training take place later on.
3. Lexicalisation:
   The unlexicalised model was turned into a lexicalised model by
   - setting the probabilities of the lexicalised rule probabilities to the values of the respective unlexicalised probabilities
   - initialising the lexical choice and lexicalised start probabilities uniformly.

4. Lexicalised training:
   Three training iterations were performed on the training corpus, re-estimating the frequencies after each iteration.
   Comparative numbers of iterations (up to 40 iterations) showed that more iterations of lexicalised training did not have further effect on the model.

To achieve a reduction of parameters and improve the lexical choice model, we utilised the pooling option as described in section 12.3.2: all active, passive and non-finite verb frames were pooled according to shared arguments, disregarding the saturation state of the frames, in order to generalise over their arguments without taking into account their positional facilities. In addition, each of the categories describing noun phrases, noun chunks, the noun bar level and proper names was pooled disregarding the features for gender, case and number, thus allowing to generalise over open class categories like adjectives which combine with nouns disregarding these features.

12.4.2 Probability Model Evaluation

As mentioned above, main background for the development of the training strategy were the perplexity of the model as the measure of mathematical evaluation on the one hand, and the parsing accuracy of grammatical structures as the measure of linguistic evaluation on the other hand. Figure 8 displays the development of the perplexity on the training data, Figure 9 the development of the perplexity on the test data, both referring to the experiment described in section 12.4.1, illustrating lexicalised training up to its fifth iteration. As the figures show, both the perplexity on the training data and the perplexity on the test data monotonously decrease during training, which means that according to perplexity the model improves steadily and has not reached the status of over-training yet.

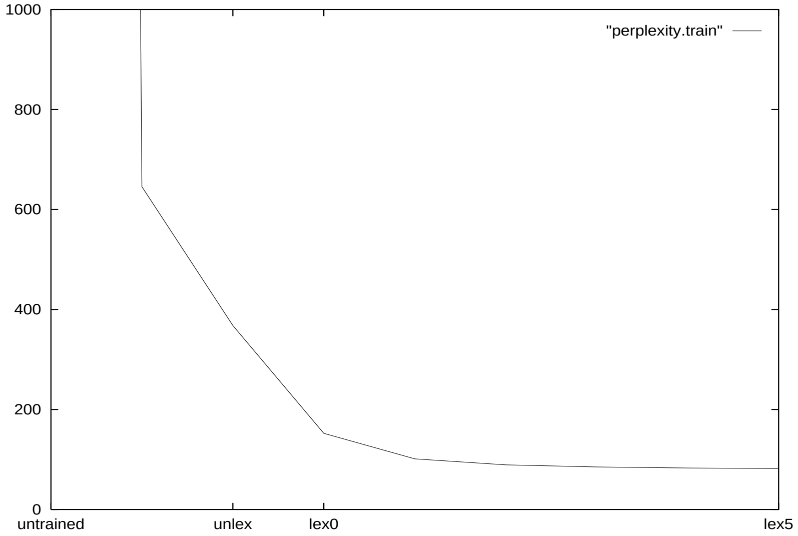


FIGURE 8 Perplexity on Training Data

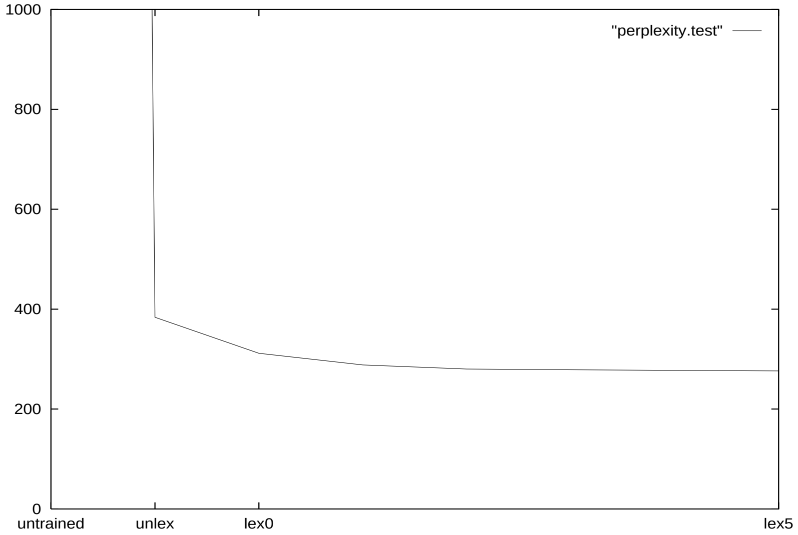


FIGURE 9 Perplexity on Test Data

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

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

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

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

with tp counting the cases where the identified chunk/label is correct, and fp counting the cases where the identified chunk/label is not correct.

Figures 10 and 11 present the strongly different development of noun chunk and subcategorisation frame representations within the models, ranging from the untrained model until the fifth iteration of lexicalised training. Noun chunks were modelled sufficiently by an unlexicalised trained grammar, lexicalisation made the modelling worse. Verb phrases in general needed a combination of unlexicalised and lexicalised training, but the representation strongly depended on the specific item. Unlexicalised training advanced frequent phenomena (compare, for example, the representation of the transitive frame with direct object for *erfahren* and with indirect object for *folgen*), lexicalisation and lexicalised training improved the lexicalised properties of the verbs, as expected.

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

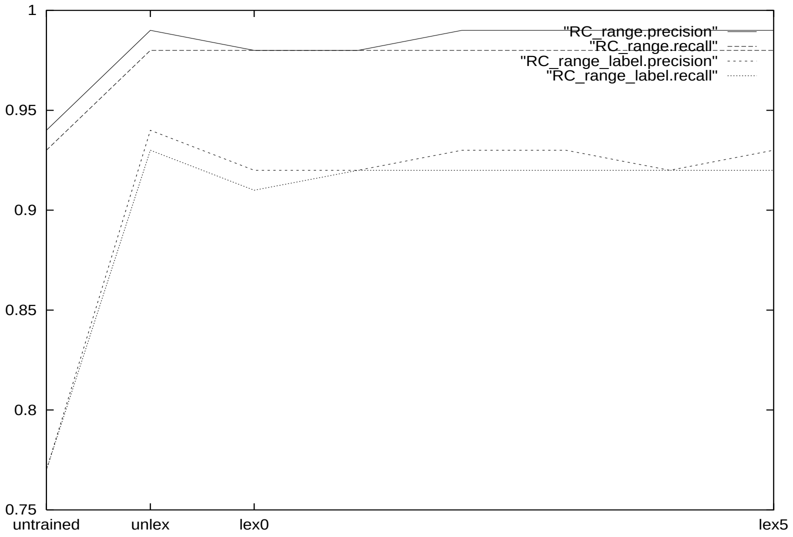


FIGURE 10 Development of Precision and Recall Values on Noun Chunk Range and Label

The precision values of the "best" model according to the training strategy in section 12.4.1 were as in Table 2.

Noun Chunks		Subcategorisation Frames on Sub-Corpora	
range	range+label	relative clauses	verb final clauses
98%	92%	63%	73%

Subcategorisation Frames on Specific Verbs					
beteiligen ‘participate’	*erhalten* ‘receive’	*folgen* ‘follow’	*verbieten* ‘forbid’	*versprechen* ‘promise’	*versuchen* ‘try’
48%	61%	88%	59%	80%	49%

TABLE 2 Precision Values on Noun Chunks and Subcategorisation Frames

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

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

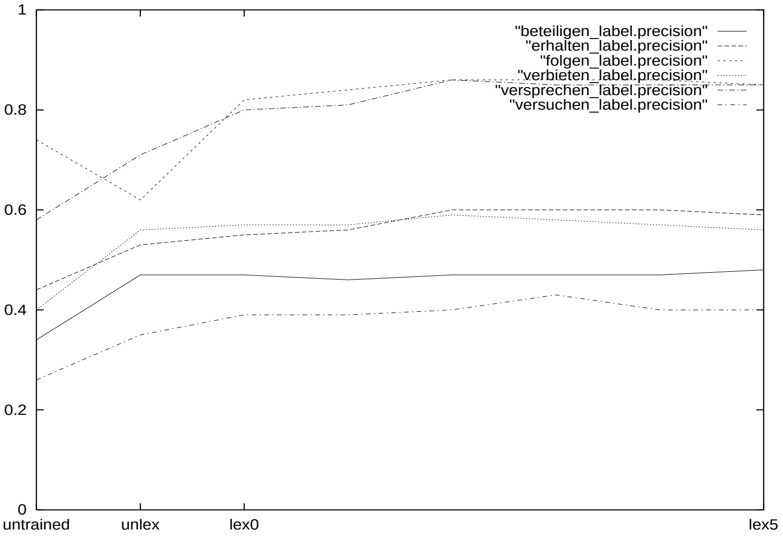


FIGURE 11 Development of Precision Values on Subcategorisation Frames for Specific Verbs

12.4.3 Investigating the Linguistic Performance of the Model

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

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

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

- Highly common subcategorisation types such as the transitive frame are learned in unlexicalised training and then slightly unlearned in lexicalised training. Less common subcategorisation types such as the demand for an indirect object are unlearned in unlexicalised training, but improved during lexicalised training.
- It is difficult and was not effectively learned to distinguish between prepositional phrases as verbal complements and adjuncts.
- The active present perfect verb complexes and passive of condition were confused, because both are composed by a past participle and a form of *to be*, e.g. *geschwommen ist* 'has swum' vs. *gebunden ist* 'is bound'.
- Copula constructions and passive of condition were confused, again because both may be composed by a past participle and a form of *to be*, e.g. *verboten ist* 'is forbidden' vs. *erfahren ist* 'is experienced'.
- Noun chunks belonging to a subcategorised non-finite clause were partly parsed as arguments of the main verb. For example, *der ihn zu überreden versucht* 'who him$_{acc}$ tried to persuade' was parsed as demanding an accusative plus a non-finite clause instead of recognising that the accusative object is subcategorised by the embedded infinitival verb.
- Reflexive pronouns appeared in the subcategorisation frame as either reflexive pronoun itself or as accusative or dative noun chunk. The correct or wrong choice of frame type containing the reflexive pronoun was learned consequently right or wrong for different verbs. For example, the verb *sich befinden* 'to be situated' was generally parsed as a transitive, not as inherent reflexive verb.

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

- The verb *beteiligen* was mostly parsed as transitive verb. Two sources of mistakes were combined here: (i) the verb was assigned a transitive instead of inherent reflexive frame, and (ii) the obligatory prepositional phrase was consequently parsed as adjunct instead of argument. All feature tendencies were already determined by unlexicalised training and not corrected in lexicalised training.
- The transitive frame of *erhalten* was recognised well, not many mistakes were made except for the PP-assignment.
- As consequence of unlexicalised training, the verb *folgen* was partly parsed as transitive, but lexicalised training corrected that tendency.
- The main problem for the verb *verbieten* was being assigned a copula-construction instead of a passive of condition.
- For the verb *versprechen* the main mistake was using the dominance of the bitransitive frame also for parsing the transitive reflexive verb *sich versprechen*.
- The main mistake for *versuchen* was parsing a direct object instead of recognising the object's correlation with the embedded infinitival verb.

We conclude the linguistic feature description by presenting probability distributions of selected verbs over subcategorisation frames in Table 3[12], as extracted by questioning tools on the model parameters.

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

Verb	Prob.	Frame	Example
erlauben	42%	IP.na	*weil meine Eltern vieles erlaubt haben* ‘because my parents allowed a lot’
	29%	IP.nad	*weil sie mir vieles erlaubt haben* ‘because they allowed me a lot’
achten	45%	IP.np	*weil das Kind auf die Ampel achten sollte* ’because the child should pay attention to the traffic lights’
	31%	IP.na	*daß wir die Bemühungen achten* ‘that we respect the effort’
	19%	IP.n	[intransitive use not possible]
basieren	89%	IP.np	*daß die Ausnahme auf der Regel basiert* ‘that the exception is based on the rule’
erfahren	50%	IP.na	*weil er die Neuigkeit erfahren hat* ’because he found out the news’
	25%	IP.np	*weil er von den Änderungen erfahren will* ’since he wants to find out about the changes’
	11%	IP.n	[intransitive use not possible]
	10%	IP.nap	[PP cannot be argument]
folgen	67%	IP.nd	*weil er ihr folgen wollte* ‘because he wanted to follow her’
	13%	IP.n	*weil wichtige Entscheidungen folgen* ‘because important decisions follow’
beginnen	48%	IP.np	*daß wir mit der Schule beginnen möchten* ‘that we want to start with school’
	24%	IP.n	*daß die Vorlesung beginnt* ‘that the seminar starts’
	11%	IP.na	*weil wir das Spiel bereits begonnen haben* ‘because we started the game already’
scheinen	32%	IP.ni	*weil die Regelung zu funktionieren scheint* ‘because the regulation seems to work’
	25%	IP.n	*weil die Sonne heute scheint* ‘because the sun is shining today’
	16%	IP.nai	[accusative should be parsed as direct object of embedded infinitival verb]
erweisen	61%	IP.nr	[PP as argument needed]
	17%	IP.npr	*weil sie sich als eine gute Fee erwiesen hat* ‘because she proved to be a fairy’
	11%	IP.nad	*weil er ihr die Ehre erweist* ‘because he paid her respect’
enden	66%	IP.np	*weil die Stunde um 9 Uhr endet* ‘because the lesson ends at 9 a.m.’
	29%	IP.n	*weil auch die besten Zeiten enden werden* ‘because even the best times will end’
beteiligen	48%	IP.npr	*weil wir uns an dem Kauf beteiligen* ‘since we participate in the purchase’
	22%	IP.np	[confusion copula construction / passive of condition]
	15%	IP.nr	[PP as argument needed]

TABLE 3 Probability Distribution over Subcategorisation Frames

12.5 Exploiting the Lexicalised Probabilistic Grammar Model

Having trained the statistical grammar models, we are equipped with valuable lexical information. But how to detect it? What are the possibilities to determine relevant lexical information and apply it to interesting tasks? The following sections refer to the potential of the grammar models, with section 12.5.1 presenting a collection of lexicalised probabilities for verbs; section 12.5.2 applies Viterbi parsing on basis of the lexical probabilities to an example sentence, followed by section 12.5.3 extracting an empirical database of subcategorisation frames from Viterbi parses; finally, section 12.5.4 explains how to base a chunker on the trained grammar.

12.5.1 Lexicalised Probabilities

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

```
Verb: glauben ‘believe’               Verb: geben ‘give’
------------------------------------  -----------------------------------
prob        IP.<frame>                prob        IP.<frame>
------------------------------------  -----------------------------------
0.45115     IP.n                      0.51598     IP.na
0.14787     IP.na                     0.22681     IP.nap
0.13740     IP.np                     0.15378     IP.nad

Verb: folgen ‘follow’                 Verb: enden ‘end’
------------------------------------  -----------------------------------
prob        IP.<frame>                prob        IP.<frame>
------------------------------------  -----------------------------------
0.70054     IP.nd                     0.66980     IP.np
0.13717     IP.n                      0.28282     IP.n

Verb: achten ‘respect/pay attention’  Verb: beteiligen ‘participate’
------------------------------------  -----------------------------------
prob        IP.<frame>                prob        IP.<frame>
------------------------------------  -----------------------------------
0.45376     IP.np                     0.52067     IP.npr
0.30238     IP.na                     0.18734     IP.np
0.18469     IP.n                      0.14666     IP.nr
```

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

```
Verb: sprechen ‘talk’   VP: VPA.np
-------------------------------------------------------
prob        rule
-------------------------------------------------------
0.18752     PP.Dat:von   ‘about’
0.13271     PP.Akk:für   ‘for’
0.13136     PP.Dat:mit   ‘with’

Verb: enden ‘end’   VP: VPA.np
-------------------------------------------------------
prob        rule
-------------------------------------------------------
0.25152     PP.Dat:mit   ‘with’
0.22102     PP.Dat:in    ‘in’
0.20671     PP.Dat:an    ‘at’

Verb: eignen ‘qualify’   VP: VPA.npr
-------------------------------------------------------
prob        rule
-------------------------------------------------------
0.39232     PP.Akk:für   ‘for’
0.15285     PP.Dat:zu    ‘to’
```

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

```
Verb: entstammen ‘descend from’   VP: VPA.nd -- NP.Dat
-------------------------------------------------------
freq      word
-------------------------------------------------------
3.0       Familie            ‘family’
3.0       Jahrhundert        ‘century’
3.0       Welt               ‘world’
2.0       Disziplin          ‘discipline’
2.0       Drogenhandel       ‘drug trafficking’
2.0       Elternhaus         ‘(parental) home’
2.0       Zeit               ‘time’

Verb: drohen ‘threaten’   VP: VPA.nd -- NP.Nom
-------------------------------------------------------
freq      word
-------------------------------------------------------
18.9      Gefahr             ‘danger’
17.0      Abschiebung        ‘deportation’
17.0      Verfolgung         ‘prosecution’
```

```
13.8      Todesstrafe         ‘death penalty’
 7.9      Tod                 ‘death’
 5.0      Arbeitslosigkeit    ‘unemployment’
 5.0      Ausweisung          ‘instruction’
 5.0      Entlassung          ‘dismissal’
 5.0      Kündigung           ‘termination’

Verb: erziehen ‘educate’   VP: VPA.na -- NP.Akk
-------------------------------------------------------
freq      word
-------------------------------------------------------
16.0      Kind                ‘child’
 2.0      Junge               ‘boy’
 2.0      Sohn                ‘son’
 2.0      Tochter             ‘daughter’
```

12.5.2 Viterbi Parses

With LoPar, it is possible to parse a corpus unambiguously by selecting the respective analysis with the highest probability (called *Viterbi parse*). Viterbi parses are printed in a list notation; graphical tools allow the parse tree representation. For example, the Viterbi parse of the relative clause *die vielen Menschen das Leben retten könnte* ‘which could save many people’s lives’ is represented by the parse tree in Figure 12. The parser correctly chose the ditransitive subcategorisation frame `nad` for the verb *retten* ‘save’, and provided the relevant `NP`s with the correct case, *die* as a nominative relative pronoun, *vielen Menschen* as an `NP` with dative case, and *das Leben* as an `NP` with accusative case. Viterbi parsing is used to build large parsed corpora (called *treebanks*), or as an intermediate step in larger NLP systems for e.g. machine translation, text mining, information retrieval, question answering, query analysis.

12.5.3 Empirical Subcategorisation Frame Database

Section 12.5.2 introduced Viterbi parses as a method for determining the most probable parse of a sentence. We collected the parses to build an empirical database, an input to complex NLP systems. The database has actually been used for semantic clustering (cf. (Rooth et al. 1999, Schulte im Walde 2000a)) and experiments on verb biases concerning lexical syntactic preferences (Lapata et al. To appear).

For example, the following lines represent some example subcategorisation frames tokens for English, extracted from the Viterbi parses of the respective sentences in the *British National Corpus (BNC)*. Each line represents one subcategorisation frame; the verb as well as the arguments are defined by a 2-/3-/4-tuple describing the syntactic category and its features: each syntactic category was accompanied by the lexical head, the prepositional phrase by the lexical head plus the head noun of

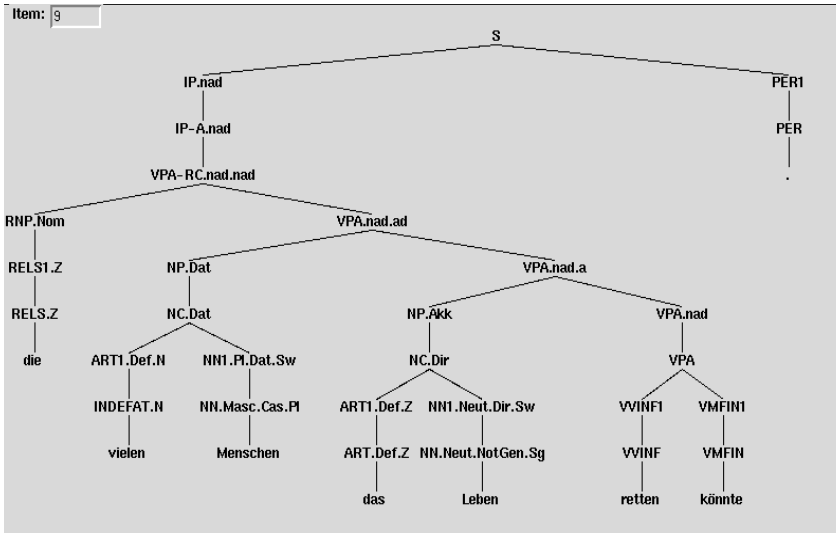


FIGURE 12 Viterbi Parse

the sub-ordinated noun phrase, and the verb by its mode.
The frames start with the description of the verb, followed by all arguments, in the order they appeared in the parses. To give an example, the frame token

```
act*excelled subj*nobody obj*him pp*in*judgement
```

describes the sentence *Nobody excelled him in that judgement.*

```
pas*described obj*realism pp*by*pn*fischer
act*proved subj*distinction ap*difficult
act*took subj*this obj*forms
act*argued subj*he pp*against*type
act*intend subj*museum to*act*sponsor
pas*limited obj*writing pp*by*demands
act*has subj*critic obj*advantage
act*serve subj*comparison obj*us pp*as*example
act*seem subj*they to*act*proceed
act*demands subj*pn*michelangelo obj*preference
```

A more detailed description of the frame tokens can be found in (Schulte im Walde 1998).

A comparable database was created for German. Following are examples starting with a verb-final clause, followed by all arguments and the verb frame.

```
S       dass in diesem Jahr der grosse Coup gelingen würde
        ‘that the big coup would succeed this year’
NP.Nom  Coup
IP.n    gelingen

S       weil die Stadtväter Schmiergelder für die Einrichtung
         eines modernen Müllplatzes einsteckten
        ‘because the city management accepted bribe money
         for the establishment of a modern dump’
NP.Nom  Stadtväter
NP.Akk  Schmiergelder
IP.na   einsteckten

S       dass diese Kunst menschlichen Bedürfnissen entspricht
        ‘that this art corresponds to human needs’
NP.Nom  Kunst
NP.Dat  Bedürfnissen
IP.nd   entspricht
```

12.5.4 Chunking

A *chunker* is a tool which marks all –possibly recursive– chunks in a sentence. Arbitrary syntactic categories can be defined as relevant chunks. Whereas the context-free grammars under development often cope with restricted parts of the respective language (cf. the German grammar described in section 12.2), we developed a language-independent method which allows to extend the grammars with robustness rules, to extract various kinds of chunks from unrestricted text.

The best chunk sequence of a sentence is defined as the sequence of chunks (with category, start and end position) for which the sum of the probabilities of all parses which contain exactly that chunk sequence is maximal. The algorithm sums probabilities up to the level of the chunks like the Inside algorithm and computes the maximum above the level of chunks like the Viterbi algorithm. To be more specific, we compute for each node n in the parse forest

- the maximum of the probabilities of all analyses of n containing chunks, and
- the sum of the probabilities of all analyses of n containing no chunks.

We have concentrated the chunking on nouns (cf. (Schmid and Schulte im Walde 2000)), since many low-level NLP systems are using them, e.g. as index terms in information retrieval or as candidates for terminology extraction.

The German base grammar currently covering verb final and relative clauses has automatically been extended by robustness rules. All rules have been trained on unlabelled data by the probabilistic context-free parser. For extracting noun chunks, the parser generates all possible noun chunk analyses, scores them and chooses the most probable chunk sequences according to the above algorithm. LoPar is able to generate chunked output in which either minimal (i.e. non-recursive) chunks or maximal chunks are marked with surrounding brackets.

The following example presents a German sentence, followed by the noun chunks extracted. The noun chunks are marked by case.

```
S       Damit sei freilich noch keine Garantie gegeben,
        schreiben beide Politiker weiter,
        dass die Verhandlungen tatsächlich während des Gipfeltreffens
        in Amsterdam zu einem guten Ende gelangten.
        ‘There is still no warranty,
         the politicians continued,
         that the negotiations at the summit meeting
         im Amsterdam conclude with a good solution.’
NC.Nom  keine Garantie
NC.Nom  beide Politiker
NC.Nom  die Verhandlungen
NC.Gen  des Gipfeltreffens
NC.Dat  Amsterdam
NC.Dat  einem guten Ende
```

12.6 Lexical Semantic Clusters

This section presents a method for automatic induction of semantically annotated subcategorisation frames from unannotated corpora. We use the statistical system for inducing subcategorisation frames for verbs as described in section 12.5.3, which estimates probability distributions and corpus frequencies for pairs of a verbal head and a subcategorisation frame. Since the statistical parser can also collect frequencies for the nominal fillers of slots in a subcategorisation frame, the induction of labels for slots in a frame is based upon the estimation of a probability distribution over tuples consisting of a class label, a selecting head, a grammatical relation, and a filler head. The class label is treated as hidden data in the EM-framework for statistical estimation. For further information on theory and applications of our clustering model see (Rooth et al. 1998) and (Rooth et al. 1999).

12.6.1 EM-Based Clustering

Basic Idea

In our clustering approach, classes are derived directly from distributional data—a sample of pairs of verbs and nouns, gathered by parsing an unannotated corpus and extracting the fillers of grammatical relations. Semantic classes corresponding to such pairs are viewed as hidden variables or unobserved data in the context of maximum likelihood estimation from incomplete data via the EM algorithm. This approach allows us to work in a mathematically well-defined framework of statistical inference, i.e., standard monotonicity and convergence results for the EM algorithm extend to our method.

The basic ideas of our EM-based clustering approach were presented in (Rooth 1995) (see also (Rooth 1998)). An important property of our clustering approach is the fact that it is a "soft" clustering method, defining class membership as a conditional probability distribution over verbs and nouns. In contrast, in hard (Boolean) clustering methods such as that of (Brown et al. 1992), every word belongs to exactly one class, which because of homophony is unrealistic. The foundation of our clustering model upon a probability model furthermore contrasts with the merely heuristic and empirical justification of similarity-based approaches to clustering (Dagan et al. 1999). The probability model we use can be found earlier in (Pereira et al. 1993). However, in contrast to this approach, our statistical inference method for clustering is formalised clearly as an EM-algorithm. Approaches to probabilistic clustering similar to ours were presented recently in (Saul and Pereira 1997) and (Hofmann and Puzicha 1998). There also EM-algorithms for similar probability models have been derived, but applied only to simpler tasks not involving a combination of EM-based clustering models as in our lexicon induction experiment.

General Theory

We seek to derive a joint distribution of verb-noun pairs from a large sample of pairs of verbs $v \in V$ and nouns $n \in N$. The key idea is to view v and n as conditioned on a hidden class $c \in C$, where the classes are given no prior interpretation. The semantically smoothed probability of a pair (v, n) is defined to be:

$$p(v,n) = \sum_{c \in C} p(c,v,n) = \sum_{c \in C} p(c)p(v|c)p(n|c)$$

The joint distribution $p(c, v, n)$ is defined by $p(c, v, n) = p(c)p(v|c)p(n|c)$. Note that by construction, conditioning of v and n on each other is solely made through the classes c.

In the framework of the EM algorithm (Dempster et al. 1977, McLachlan and Krishnan 1997), we can formalise clustering as an estimation problem for a latent class (LC) model as follows. We are given:

- a sample space $\mathcal{Y}$ of observed, incomplete data, corresponding to pairs from $V \times N$,
- a sample space $\mathcal{X}$ of unobserved, complete data, corresponding to triples from $C \times V \times N$,

- a set $X(y) = \{x \in \mathcal{X} \mid x = (c, y),\ c \in C\}$ of complete data related to the observation y,
- a complete-data specification $p_\theta(x)$, corresponding to the joint probability $p(c, v, n)$ over $C \times V \times N$, with parameter-vector $\theta = \langle \theta_c, \theta_{vc}, \theta_{nc} | c \in C, v \in V, n \in N \rangle$,
- an incomplete data specification $p_\theta(y)$ which is related to the complete-data specification as the marginal probability $p_\theta(y) = \sum_{X(y)} p_\theta(x)$.

The EM algorithm is directed at finding a value $\hat{\theta}$ of θ that maximises the incomplete-data log-likelihood function L as a function of θ for a given sample $\mathcal{Y}$, i.e.,

$$\hat{\theta} = \arg\max_\theta L(\theta) \text{ where } L(\theta) = \ln \prod_y p_\theta(y).$$

As prescribed by the EM algorithm, the parameters of $L(\theta)$ are estimated indirectly by proceeding iteratively in terms of complete-data estimation for the auxiliary function $Q(\theta; \theta^{(t)})$, which is the conditional expectation of the complete-data log-likelihood $\ln p_\theta(x)$ given the observed data y and the current fit of the parameter values $\theta^{(t)}$ (E-step). This auxiliary function is iteratively maximised as a function of θ (M-step), where each iteration is defined by the map

$$\theta^{(t+1)} = M(\theta^{(t)}) = \arg\max_\theta Q(\theta; \theta^{(t)})$$

Note that our application is an instance of the EM-algorithm for context-free models (Baum et al. 1970, Baker 1979), from which the following particularly simple re-estimation formulae can be derived. Let $x = (c, y)$ for fixed c and y, and $f(y)$ be the frequency of y in the training sample. Then

$$\begin{aligned} M(\theta_{vc}) &= \frac{\sum_{y \in \{v\} \times N} f(y) p_\theta(x|y)}{\sum_y f(y) p_\theta(x|y)}, \\ M(\theta_{nc}) &= \frac{\sum_{y \in V \times \{n\}} f(y) p_\theta(x|y)}{\sum_y f(y) p_\theta(x|y)}, \\ M(\theta_c) &= \frac{\sum_y f(y) p_\theta(x|y)}{|\mathcal{Y}|}. \end{aligned}$$

Intuitively, the conditional expectation of the number of times a particular v, n, or c choice is made during the derivation is prorated by the conditionally expected total number of times a choice of the same kind is made. As shown by (Baum et al. 1970), every such maximisation step increases the log-likelihood function L, and a sequence of re-estimates eventually converges to a (local) maximum of L.

Clustering Examples

In the following, we will present some examples of induced clusters. In one experiment the input to the clustering algorithm was a training corpus of 1,178,698 tokens (608,850 types) of English verb-noun pairs participating in the grammatical relations of intransitive and transitive verbs and their subject and object fillers. The data were gathered from the maximal-probability parses the head-lexicalised probabilistic context-free grammar of (Carroll and Rooth 1998) gave for the British National Corpus (117 million words).

Figure 13 shows an induced semantic class out of a model with 35 classes. At the top are listed the 30 most probable nouns in the $p(n|5)$ distribution and their probabilities, and at left are the 30 most probable verbs in the $p(v|5)$ distribution where 5 is the class index. Those verb-noun pairs which were seen in the training data appear with a dot in the class matrix. Verbs with suffix $.as:s$ indicate the subject slot of an active intransitive. Similarly $.aso:s$ denotes the subject slot of an active transitive, and $.aso:o$ denotes the object slot of an active transitive. Thus v in the above discussion actually consists of a combination of a verb with a subcategorisation frame slot $as:s$, $aso:s$, or $aso:o$.

Induced classes often have a basis in lexical semantics; class 5 can be interpreted as clustering agents, denoted by proper names, 'man', and 'woman', together with verbs denoting *communicative action*. Figure 14 shows a cluster involving verbs of *scalar change* and things which can move along scales. Figure 15 can be interpreted as involving different *dispositions* and modes of their execution.

In another experiment, we extracted 418,290 tokens (318,086 types) of pairs of German verbs or adjectives and grammatically related nouns from maximal-probability parses; the parsed corpus was the verb final sub-corpus from the HGC described in section 12.2.1. The underlying lexicalised statistical model for German was described in section 12.4.

Figure 16 and Figure 17 show two classes out of a model with 35 classes. On the left and at the top are listed the 30 highest probable verb/adjective predicates and nouns appearing as fillers of the verb/adjective slots, ordered according to their probability given the class. Verbal predicates are annotated with subcategorisation slots, e.g., *liegen-(A)np:n* denotes the nominative noun-phrase filler (n) of the subject-slot of an active verb (A) *liegen* 'lie' subcategorising for a nominative (n) and a prepositional (p) phrase. *tragen-(A)na:a* is the accusative noun-phrase filler (a) of the object slot of the transitive verb *tragen* 'carry', *steigen-(A)n:n* denotes the nominative filler (n) of the subject slot of the intransitive verb *steigen* 'rise'. Clearly, due to the

Cl. 5 PROB: 0.0412		0.0148	0.0084	0.0082	0.0078	0.0074	0.0071	0.0054	0.0049	0.0048	0.0047	0.0046	0.0041	0.0040	0.0040	0.0039	0.0039	0.0039	0.0039	0.0038	0.0038	0.0038	0.0037	0.0035	0.0035	0.0035	0.0034	0.0033	0.0033	0.0033	0.0033
		man	ruth	corbett	doctor	woman	athelstan	cranston	benjamin	stephen	adam	girl	laura	maggie	voice	john	harry	emily	one	people	boy	rachel	ashley	jane	caroline	jack	burun	juliet	blanche	helen	edward
0.0542	ask.as:s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●
0.0340	nod.as:s		●	●	●	●	●	●	●	●		●	●	●		●	●	●			●	●	●	●	●	●	●	●	●	●	●
0.0299	think.as:s	●	●	●	●	●	●			●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●		●	●	●	●
0.0287	shake.aso:s	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●		●	●	●	●	●	●	●	●	●		●	●
0.0264	smile.as:s	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●		●	●	●	●	●
0.0213	laugh.as:s	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
0.0207	reply.as:s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●	●
0.0167	shrug.as:s	●		●	●	●	●	●	●	●	●	●	●	●			●	●			●	●		●	●	●	●	●	●	●	●
0.0148	wonder.as:s	●	●	●	●	●	●			●	●	●				●	●	●		●	●	●	●	●	●	●	●	●	●	●	●
0.0141	feel.aso:s	●		●	●	●				●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●		●	●	●	●
0.0133	take.aso:s	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●
0.0121	sigh.as:s	●	●	●	●	●	●		●	●	●		●			●	●	●		●		●	●	●	●	●	●		●	●	●
0.0110	watch.aso:s	●	●	●		●	●	●	●	●	●	●		●		●	●	●	●	●	●	●		●		●	●	●		●	
0.0106	ask.aso:s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●			●		●			●	●	●
0.0104	tell.aso:s	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●
0.0094	look.as:s	●	●	●	●	●	●	●	●	●	●	●				●		●	●	●	●	●				●	●	●			●
0.0092	give.aso:s	●	●		●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
0.0089	hear.aso:s	●	●	●	●	●	●	●				●		●	●	●	●		●	●	●			●	●	●	●	●	●		
0.0083	grin.as:s	●	●	●	●	●	●	●	●		●	●	●	●		●	●	●			●	●	●	●	●	●	●			●	●
0.0083	answer.as:s	●	●	●	●	●	●	●	●					●	●	●	●	●	●	●	●		●		●	●		●		●	●
0.0081	explain.as:s		●		●	●			●	●	●		●	●	●	●	●	●	●		●	●	●	●				●	●	●	●
0.0079	frown.as:s	●	●		●	●				●	●	●	●				●	●	●	●	●	●	●	●	●					●	●
0.0076	hesitate.as:s	●	●		●	●				●	●		●		●	●	●	●	●		●	●	●		●	●		●			●
0.0074	stand.as:s	●	●	●	●	●	●	●		●		●	●		●	●	●		●	●	●	●	●		●	●		●			
0.0066	continue.as:s	●	●	●	●		●	●	●	●	●	●	●		●	●	●	●	●	●	●			●				●	●	●	
0.0065	find.aso:s	●	●		●	●			●		●	●	●	●		●	●	●	●	●	●	●		●		●		●	●		●
0.0064	feel.as:s	●	●	●	●	●		●	●		●	●		●			●	●	●	●	●	●	●	●	●	●		●	●		●
0.0062	sit.as:s	●	●	●	●	●	●	●	●			●	●	●		●	●	●	●	●	●	●		●				●	●		●
0.0062	agree.as:s	●	●	●	●		●		●	●	●	●	●	●		●		●	●	●	●	●	●	●		●	●	●	●		●
0.0056	cry.as:s	●	●	●		●		●		●	●	●	●		●				●	●	●	●		●		●					

FIGURE 13 English Class 5: *communicative action*

smaller size of the German input data compared to the English data, German classes are less dense than the English counterparts.

Figure 16 shows a cluster involving scalar motion verbs and things which can move along scales. Figure 17 shows a class which can be interpreted as *governmental/public authority*, involving nouns such as *police force* and *public prosecutor's office*.

Cl. 17 PROB: 0.0265		0.0379	0.0315	0.0313	0.0249	0.0164	0.0143	0.0110	0.0109	0.0105	0.0103	0.0099	0.0091	0.0089	0.0088	0.0082	0.0077	0.0073	0.0071	0.0071	0.0070	0.0068	0.0067	0.0065	0.0065	0.0058	0.0057	0.0057	0.0054	0.0051	0.0050
		number	rate	price	cost	level	amount	sale	value	interest	demand	chance	standard	share	risk	profit	pressure	income	performance	benefit	size	population	proportion	temperature	tax	fee	time	power	quality	supplely	money
0.0437	increase.as:s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
0.0392	increase.aso:o		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●	●	●	●
0.0344	fall.as:s	●	●	●	●	●	●	●	●		●	●	●	●		●	●	●	●			●	●	●	●		●		●	●	
0.0337	pay.aso:o	●	●	●		●	●		●	●	●			●				●	●	●			●			●	●				●
0.0329	reduce.aso:o	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
0.0257	rise.as:s	●	●	●	●	●	●	●	●		●		●	●	●	●	●	●		●	●	●	●	●	●		●	●	●	●	
0.0196	exceed.aso:o	●	●	●	●	●	●		●	●	●	●	●			●		●		●	●	●	●		●	●	●	●	●	●	
0.0177	exceed.aso:s	●	●	●	●	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●		●			●	●
0.0169	affect.aso:o	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●		●	●	●	●	●	
0.0156	grow.as:s	●						●	●	●	●			●	●	●	●	●		●	●	●	●					●			●
0.0134	include.aso:s	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●		●	●			●	●	●	●	●	●
0.0129	reach.aso:s	●	●	●	●	●	●	●		●	●			●		●	●		●	●		●	●	●			●	●	●	●	●
0.0120	decline.as:s	●	●	●	●	●		●	●	●	●		●	●	●	●					●	●	●	●					●	●	
0.0102	lose.aso:o	●	●		●		●	●	●	●		●	●	●	●	●	●	●		●		●	●	●		●	●	●	●		●
0.0099	act.aso:s		●	●				●					●							●					●			●		●	
0.0099	improve.aso:o	●	●			●			●			●	●	●		●			●								●	●	●		
0.0088	include.aso:o	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
0.0088	cut.aso:o	●	●	●	●	●		●	●			●	●		●	●		●		●					●	●	●	●			
0.0080	show.aso:o	●	●		●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●				●	●	●		
0.0078	vary.as:s	●	●	●	●	●	●		●	●	●		●						●		●		●	●			●		●		
0.0072	give.aso:o	●	●	●	●	●	●		●	●		●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●		●
0.0071	carry.aso:o	●	●	●	●	●	●			●			●		●			●		●		●	●		●	●		●	●		●
0.0068	improve.as:s	●	●	●		●		●	●			●	●			●			●								●	●	●		
0.0066	have.as:s	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●		●	●	●	●	●	●	●
0.0066	produce.aso:o	●	●		●	●	●	●	●	●	●		●	●		●	●	●	●	●	●		●				●	●	●	●	●
0.0066	get.aso:o	●	●	●	●	●	●	●	●	●	●		●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
0.0064	raise.aso:o	●		●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●				●	●				●	●	●
0.0063	mean.aso:o	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●		●		●	●	●	●	●
0.0062	receive.aso:o	●		●	●	●	●		●	●	●	●	●	●		●		●	●	●		●			●	●	●	●	●	●	●
0.0058	stand.aso:o		●									●			●												●				●

FIGURE 14 English Class 17: *scalar change*

Cl. 8 PROB: 0.0369		0.0385	0.0162	0.0157	0.0101	0.0073	0.0071	0.0063	0.0060	0.0060	0.0057	0.0055	0.0052	0.0052	0.0051	0.0050	0.0050	0.0048	0.0047	0.0047	0.0046	0.0041	0.0041	0.0040	0.0040	0.0039	0.0038	0.0037	0.0037	0.0036	0.0036
		change	use	increase	development	growth	effect	result	degree	response	approach	reduction	forme	condition	understanding	improvement	treatment	skill	action	process	activity	knowledge	factor	level	type	reaction	kind	difference	movement	loss	amount
0.0539	require.aso:o	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●			●	●		●
0.0469	show.aso:o	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●		●	●	●	●
0.0439	need.aso:o	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●		●	●		●	●	●	●	●	●		●
0.0383	involve.aso:o	●		●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●
0.0270	produce.aso:o	●	●	●	●	●		●	●	●	●	●	●	●	●	●			●		●	●	●	●	●	●	●	●	●	●	●
0.0255	occur.as:s	●	●	●	●	●	●	●	●	●	●	●	●	●		●			●	●	●		●	●	●	●	●	●	●	●	
0.0192	cause.aso:s	●	●	●	●	●	●	●		●		●	●	●			●		●	●	●	●	●	●	●	●	●	●	●	●	●
0.0189	cause.aso:o	●	●	●	●	●	●	●	●	●				●		●			●		●			●		●	●	●	●	●	●
0.0179	affect.aso:s	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
0.0162	require.aso:s	●	●	●	●		●	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
0.0150	mean.aso:o	●	●	●	●	●	●		●		●	●	●	●	●	●	●		●	●	●	●		●	●		●	●	●	●	●
0.0140	suggest.aso:o	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●		●	●	●	●	●	●	●		●	●	●		●
0.0138	produce.aso:s	●	●	●	●	●	●	●	●	●	●	●	●	●			●		●	●	●		●	●	●	●	●	●	●	●	●
0.0109	demand.aso:o	●	●	●	●			●	●	●	●		●		●	●	●	●	●		●	●		●			●				●
0.0109	reduce.aso:s	●	●	●	●	●			●	●	●	●	●				●	●	●	●	●	●	●	●					●	●	●
0.0097	reflect.aso:o	●	●	●	●	●	●		●	●	●	●	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●
0.0092	involve.aso:s	●	●			●	●	●		●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
0.0091	undergo.aso:o	●	●	●	●		●		●					●		●	●			●						●	●			●	●
0.0086	increase.aso:s	●	●	●	●	●	●	●		●	●	●				●	●	●	●	●		●	●	●	●				●	●	●
0.0081	allow.aso:o	●	●	●	●	●			●		●	●			●	●	●		●	●				●			●		●	●	●
0.0079	include.aso:o			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●
0.0075	make.aso:s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●
0.0075	support.aso:o	●	●	●	●	●			●	●	●	●			●	●		●		●	●			●	●		●		●	●	●
0.0073	saw.aso:o	●	●	●	●	●	●	●	●		●	●	●	●	●	●			●					●	●	●	●	●	●		●
0.0072	create.aso:s	●	●	●	●	●	●	●		●	●	●	●	●		●			●	●	●		●	●	●	●	●	●	●	●	
0.0070	affect.aso:o	●	●		●	●	●	●	●	●	●	●	●	●	●				●	●	●			●	●	●	●	●	●		●
0.0069	imply.aso:o	●	●	●		●			●		●	●	●	●	●	●			●	●		●		●	●		●	●			●
0.0068	achieve.aso:o	●	●	●		●	●	●	●	●		●	●		●	●		●		●				●		●	●		●	●	●
0.0066	find.aso:o	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●
0.0062	describe.aso:o	●	●	●	●	●	●	●		●	●		●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●

FIGURE 15 English Class 8: *dispositions*

Cl. 26 PROB: 0.0223		0.0379	0.0226	0.0182	0.0171	0.0137	0.0133	0.0129	0.0086	0.0079	0.0079	0.0073	0.0072	0.0071	0.0067	0.0064	0.0063	0.0061	0.0057	0.0056	0.0053	0.0051	0.0047	0.0046	0.0039	0.0038	0.0037	0.0037	0.0036	0.0036	0.0035
		Ergebnis	Preis	Menge	Anteil	Stück	Zahl	Gewinn	Kritiker	BürgerMeister	Angst	Umsatz	Einnahme	Zins	Schicksal	Bus	NachFrage	Ertrag	Figur	Verlust	Frucht	ArbeitsLosigkeit	Kost	Last	WahlErgebnis	Temperatur	Solidarität	InflationsRate	Abschluss	Import	unterSchrift
0.0601	liegen −(A)np:n−	●	●	●	●	●	●	●		●				●	●			●		●			●			●		●			
0.0351	tragen −(A)na:a−			●	●									●						●	●		●	●							
0.0230	steigen −(A)n:n−		●	●	●		●					●	●	●			●			●		●				●		●			
0.0213	sagen −(A)n:n−								●	●																					
0.0182	gering −ADJ−		●	●	●		●						●	●				●		●		●	●			●		●			
0.0135	sinken −(A)n:n−		●		●			●					●	●			●	●								●					
0.0135	steigen −(A)np:n−				●		●										●	●		●		●				●					
0.0119	erklären −(A)n:n−									●																					
0.0100	positiv −ADJ−	●																●													
0.0091	gehen −(A)np:n−				●		●	●				●	●					●										●		●	
0.0087	sinken −(A)np:n−				●		●					●		●												●	●			●	
0.0082	spät −ADJ−												●					●													
0.0077	wirken −(A)n:n−																														
0.0071	ändern −(A)na:a−														●																
0.0071	regional −ADJ−																														
0.0067	Erfolgreich −ADJ−																												●		
0.0065	bestätigen −(A)n:n−									●	●																				
0.0057	steigen −ADJ−		●				●	●				●		●			●					●	●					●			
0.0056	an+steigen −(A)np:n−		●		●		●					●										●				●		●			
0.0054	formulieren −(A)na:n−									●																					
0.0052	böse −ADJ−														●																
0.0051	an+steigen −(A)n:n−				●			●									●					●	●			●					
0.0051	sitzen −(A)n:n−										●																				
0.0049	übersteigen −(A)na:n−	●	●		●		●					●	●	●			●			●			●					●			
0.0045	ein+setzen −(P)n:n−															●															
0.0045	an+erkennen −(A)na:a−	●																					●		●				●		
0.0045	entdecken −(A)nap:a−										●								●												
0.0043	zu+nehmen −(A)np:n−	●			●						●	●	●				●					●								●	
0.0043	betrachten −(A)na:a−						●																								
0.0043	senken −(P)n:n−		●	●			●							●			●											●			

FIGURE 16 German Class 26: *scalar change*

Cl. 14 PROB: 0.0283		0.0877	0.0303	0.0187	0.0179	0.0161	0.0154	0.0151	0.0150	0.0130	0.0112	0.0094	0.0088	0.0086	0.0085	0.0073	0.0066	0.0066	0.0064	0.0061	0.0061	0.0059	0.0058	0.0057	0.0053	0.0051	0.0051	0.0043	0.0042	0.0041	0.0041
		Polizei	deutschLand	Nation	SPD	USA	Koalition	Sprecher	Verein	StaatsAnwaltschaft	Behörde	Bonn	firm	UNO	BundesAmt	Veranstalter	Blatt	Konzern	Magistrat	Sender	oberBürgerMeister	BundesBank	BürgerMeister	fern+sehen	Nato	Zeitung	Nähe	nachrichtenAgentur	PolizeiSprecher	LandesRegierung	rundFunk
0.0581	mit+teilen −(A)n:n−	•						•	•		•				•	•		•	•	•			•					•	•		•
0.0220	berichten −(A)n:n−	•			•			•			•				•		•			•		•		•		•		•	•		•
0.0141	vereinen −ADJ−		•	•																											
0.0120	sagen −(A)n:n−	•						•								•							•								
0.0114	BremerJ −AD−	•											•							•			•							•	
0.0113	mit+teilen −(A)np:n−			•				•		•	•								•			•	•								
0.0095	erklären −(A)n:n−	•						•	•						•						•		•						•		
0.0070	hessisch −ADJ−	•			•																									•	•
0.0065	unmittelbar −ADJ−																										•				
0.0064	fordern −(A)na:n−		•		•	•			•			•							•							•				•	
0.0063	machen −(A)nad:n−	•			•	•			•	•									•												
0.0061	verzichten −(A)np:n−		•		•	•	•				•			•											•					•	
0.0060	melden −(A)n:n−							•							•									•		•		•			•
0.0059	Berliner −ADJ−	•							•	•	•									•						•				•	
0.0055	spielen −(A)nap:n−		•			•																									
0.0052	Westdeutsch −ADJ−												•					•													•
0.0046	statistisch −ADJ−														•																
0.0043	meinen −(A)n:n−					•		•											•				•								
0.0043	auf+nehmen −(A)nap:n−	•	•		•	•				•		•		•																	
0.0042	wünschen −(A)nar:a−																														
0.0041	vor+stellen −(A)nar:n−				•		•		•													•			•					•	
0.0041	haben −(A)nap:n−	•	•		•	•	•				•				•	•						•		•	•					•	
0.0041	betonen −(A)n:n−	•														•					•		•								
0.0040	zuständig −ADJ−									•	•												•								
0.0039	übernehmen −(A)na:n−	•			•	•	•		•		•	•		•				•	•												
0.0038	sächsisch −ADJ−	•			•																									•	
0.0036	örtlich −ADJ−	•							•		•															•					
0.0035	bestätigen −(A)n:n−							•		•	•										•		•						•		
0.0035	erzählen −(A)n:n−							•																							
0.0033	ein+treffen −(A)n:n−	•																								•					

FIGURE 17 German Class 14: *governmental/public authority*

12.6.2 Evaluation of Clustering Models

Pseudo-Disambiguation

We evaluated our clustering models on a pseudo-disambiguation task similar to that performed in (Pereira et al. 1993), but differing in detail. The task is to judge which of two verbs v and v' is more likely to take a given noun n as its argument where the pair (v, n) has been cut out of the original corpus and the pair (v', n) is constructed by pairing n with a randomly chosen verb v' such that the combination (v', n) is completely unseen. Thus this test evaluates how well the models generalise over unseen verbs.

The data for this test were built as follows. We constructed an evaluation corpus of (v, n, v') triples from a test corpus of 3,000 types of (v, n) pairs which were randomly cut out of the original corpus of 1,280,712 tokens, leaving a training corpus of 1,178,698 tokens. Each noun n in the test corpus was combined with a verb v' which was randomly chosen according to its frequency such that the pair (v', n) did appear neither in the training nor in the test corpus. However, the elements v, v', and n were required to be part of the training corpus. Furthermore, we restricted the verbs and nouns in the evaluation corpus to the ones which occurred at least 30 times and at most 3,000 times with some verb-functor v in the training corpus. The resulting 1,337 evaluation triples were used to evaluate a sequence of clustering models trained from the training corpus.

The clustering models we evaluated were parameterised in starting values of the training algorithm, in the number of classes of the model, and in the number of iteration steps, resulting in a sequence of $3 \times 10 \times 6$ models. Starting from a lower bound of 50% for randomly initialised models, accuracy was calculated as the number of times the model decided for $p(n|v) \geq p(n|v')$ out of all choices made. Figure 18 shows the evaluation results for models trained with 50 iterations, averaged over starting values, and plotted against class cardinality. Different starting values had an effect of ± 2% on the performance of the test. We obtained a value of about 80% accuracy for models between 25 and 100 classes. Models with more than 100 classes show a small but stable overfitting effect.

The German models were evaluated in a similar way. An evaluation corpus of 886 (v, n, v') triples was extracted from the original corpus of 428,446 verb/adjective-noun tokens, leaving 418,290 tokens for training a sequence of clustering models. Again, the models were parameterised in starting values, number of classes and iteration steps, resulting in a sequence of $3 \times 11 \times 20$ models. Figure 18 shows the evaluation results for models trained with 100 iterations, averaged over starting values, and plotted against class cardinality. We obtained an accuracy of over 75% for models up to 35 classes. Different starting values had an effect of ± 2% on the evaluation results. For models with more than 50 classes again a small overfitting effect can be seen.

Smoothing Power

A second experiment addressed the smoothing power of the model by counting the number of (v, n) pairs in the set $V \times N$ of all possible combinations of verbs and nouns which received a positive joint probability by the model. The $V \times N$-space for the above clustering models included about 425 million (v, n) combinations; we approximated the smoothing size of a model by randomly sampling 1,000 pairs from $V \times N$ and returning the percentage of positively assigned pairs in the random sample. Figure 19 plots the smoothing results for the above models against the number of classes. Starting values had an influence of ± 1% on performance. Given the proportion of the number of types in the training corpus to the $V \times N$-space, without clustering we have a smoothing power of 0.14% whereas for example a model with 50 classes and 50 iterations has a smoothing power of about 93%.

Corresponding to the maximum likelihood paradigm, the number of training iterations had a decreasing effect on the smoothing performance whereas the accuracy of the pseudo-disambiguation was increasing in the number of iterations. We found a number of 50 iterations to be a good compromise in this trade-off.

For German models we observed a baseline smoothing power of 0.012% which is the relation of the number of types in the German training corpus to the 2.5 billion combinations in the $V \times N$-space for the German experiments. Despite of the fact that this baseline is 10 times smaller than the baseline for the English models, we have a smoothing power of about 32% for models with 25 classes, which were best in terms of the pseudo-disambiguation task. This is shown in Figure 19. The best compromise in terms of iterations was a number of 100 iterations for the German experiments.

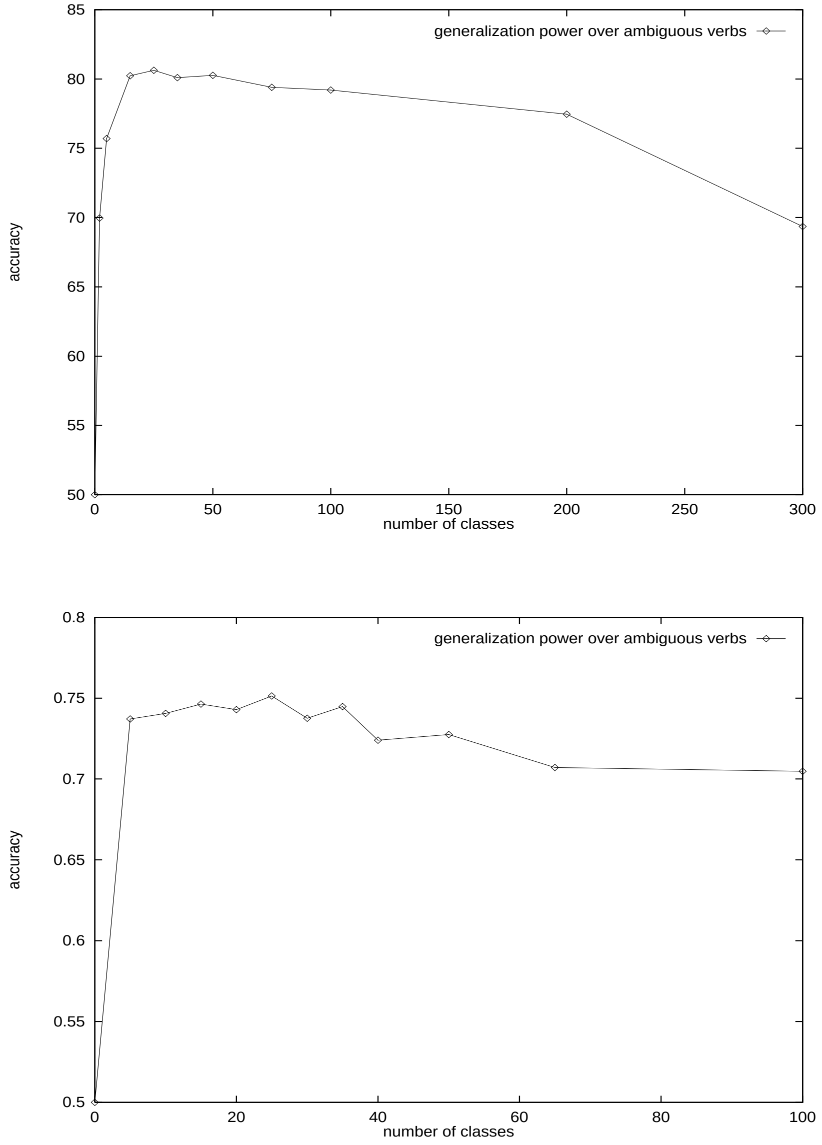


FIGURE 18 Evaluation of English/German Models on Pseudo-Disambiguation Task

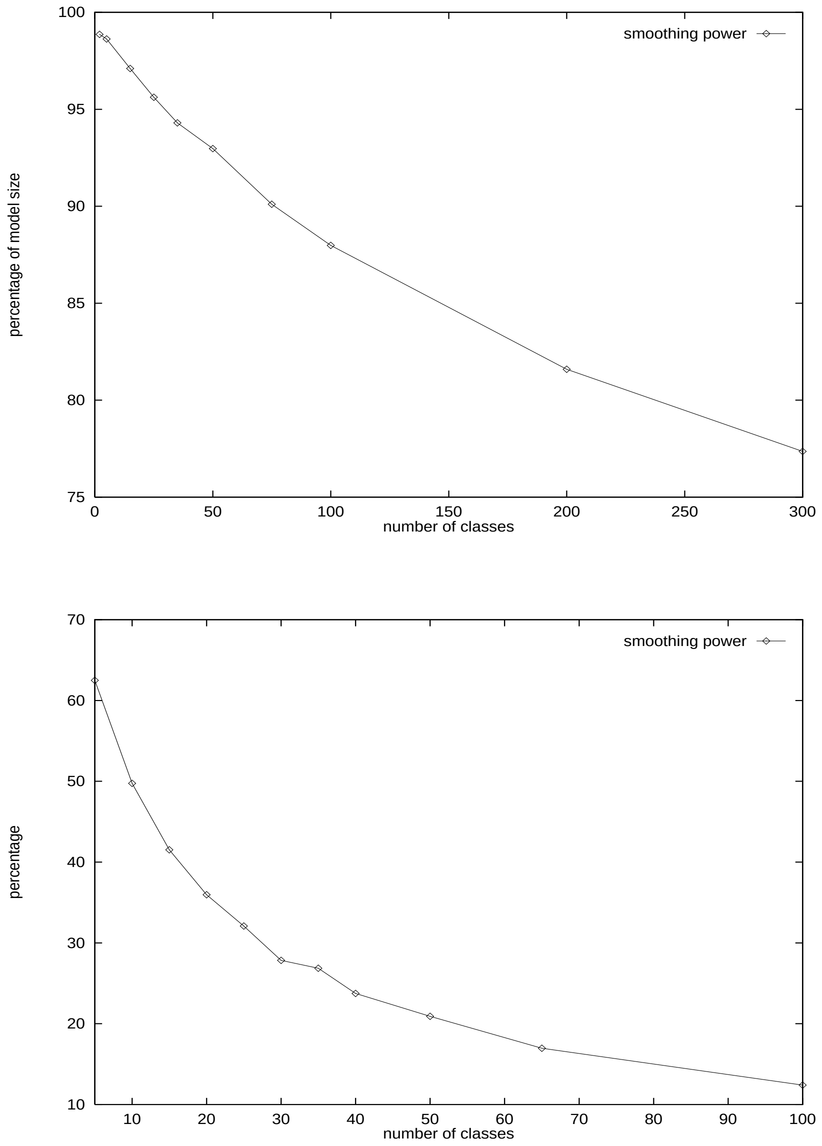


FIGURE 19 Evaluation of English/German Models on Smoothing Task

12.6.3 Lexicon Induction based on Latent Classes

The goal of the following experiment was to derive a lexicon of several hundred intransitive and transitive verbs with subcategorisation slots labelled with latent classes.

Probabilistic Labelling with Latent Classes using EM-Estimation

To induce latent classes for the subject slot of a fixed intransitive verb the following statistical inference step was performed. Given a latent class model $p_{LC}(\cdot)$ for verb-noun pairs, and a sample $n_1, \ldots, n_M$ of subjects for a fixed intransitive verb, we calculate the probability of an arbitrary subject $n \in N$ by:

$$p(n) = \sum_{c \in C} p(c,n) = \sum_{c \in C} p(c) p_{LC}(n|c).$$

The estimation of the parameter-vector $\theta = \langle \theta_c | c \in C \rangle$ can be formalised in the EM framework by viewing $p(n)$ or $p(c,n)$ as a function of θ for fixed $p_{LC}(.)$. The re-estimation formulae resulting from the incomplete data estimation for these probability functions have the following form ($f(n)$ is the frequency of n in the sample of subjects of the fixed verb):

$$M(\theta_c) = \frac{\sum_{n \in N} f(n) p_\theta(c|n)}{\sum_{n \in N} f(n)}$$

A similar EM induction process can be applied also to pairs of nouns, thus enabling induction of latent semantic annotations for transitive verb frames. Given a LC model $p_{LC}(\cdot)$ for verb-noun pairs, and a sample $(n_1, n_2)_1, \ldots, (n_1, n_2)_M$ of noun arguments (n_1 subjects, and n_2 direct objects) for a fixed transitive verb, we calculate the probability of its noun argument pairs by:

$$\begin{aligned} p(n_1, n_2) &= \sum_{c_1, c_2 \in C} p(c_1, c_2, n_1, n_2) \\ &= \sum_{c_1, c_2 \in C} p(c_1, c_2) p_{LC}(n_1|c_1) p_{LC}(n_2|c_2) \end{aligned}$$

Again, estimation of the parameter-vector $\theta = \langle \theta_{c_1 c_2} | c_1, c_2 \in C \rangle$ can be formalised in an EM framework by viewing $p(n_1, n_2)$ or $p(c_1, c_2, n_1, n_2)$ as a function of θ for fixed $p_{LC}(.)$. The re-estimation formulae resulting from this incomplete data estimation problem have the following simple form ($f(n_1, n_2)$ is the frequency of (n_1, n_2) in the sample of noun argument pairs of the fixed verb):

$$M(\theta_{c1c2}) = \frac{\sum_{n_1, n_2 \in N} f(n_1, n_2) p_\theta(c_1, c_2|n_1, n_2)}{\sum_{n_1, n_2 \in N} f(n_1, n_2)}$$

Note that the class distributions $p(c)$ and $p(c_1, c_2)$ for intransitive and transitive models can also be computed for verbs unseen in the LC model.

Lexicon Induction Experiment

In a first experiment with English data we used a model with 35 classes. From maximal probability parses for the British National Corpus derived with the statistical parser of (Carroll and Rooth 1998), we extracted frequency tables for intransitive verb/subject pairs and transitive verb/subject/object triples. The 500 most frequent verbs were selected for slot labelling. Figure 20 shows two verbs v for which the most probable class label is 5, a class which we earlier described as *communicative action*, together with the estimated frequencies of $f(n)p_\theta(c|n)$ for those ten nouns n for which this estimated frequency is highest.

blush 5	0.982975	*snarl* 5	0.962094
constance	3	mandeville	2
christina	3	jinkwa	2
willie	2.99737	man	1.99859
ronni	2	scott	1.99761
claudia	2	omalley	1.99755
gabriel	2	shamlou	1
maggie	2	angalo	1
bathsheba	2	corbett	1
sarah	2	southgate	1
girl	1.9977	ace	1

FIGURE 20 Lexicon Entries: *blush*, *snarl*

Figure 21 shows corresponding data for an intransitive scalar motion sense of *increase*.

increase 17	0.923698
number	134.147
demand	30.7322
pressure	30.5844
temperature	25.9691
cost	23.9431
proportion	23.8699
size	22.8108
rate	20.9593
level	20.7651
price	17.9996

FIGURE 21 Lexicon Entry: *increase*

Figure 22 shows the intransitive verbs which take 17 as the most probable label. Intuitively, the verbs are semantically coherent. When compared to (Levin 1993)'s 48 top-level verb classes, we found an agreement of our classification with her class of "verbs of changes of state" except for the last three verbs in the list in Figure 22 which is sorted by probability of the class label.

0.977992	decrease	0.560727	drop
0.948099	double	0.476524	grow
0.923698	increase	0.42842	vary
0.908378	decline	0.365586	improve
0.877338	rise	0.365374	climb
0.876083	soar	0.292716	flow
0.803479	fall	0.280183	cut
0.672409	slow	0.238182	mount
0.583314	diminish		

FIGURE 22 Scalar Motion Verbs

Figure 23 shows the most probable pair of classes for *increase* as a transitive verb, together with estimated frequencies for the head filler pair. Note that the object label 17 is the class found with intransitive scalar motion verbs; this correspondence is exploited in the next section.

increase $(8, 17)$	0.3097650
development - pressure	2.3055
fat - risk	2.11807
communication - awareness	2.04227
supplementation - concentration	1.98918
increase - number	1.80559

FIGURE 23 Transitive *increase* with Estimated Frequencies for Filler Pairs

Further experiments were done with two German models with 35 and 50 classes respectively. The data for these experiments were extracted from the maximal probability parses of the verb final German sub-corpus from the HGC described in section 12.2.1, parsed with the lexicalised probabilistic grammar described in section 12.4. Figure 24 shows the subjects of the transitive verb *bekanntgeben* 'make public'. The nouns are classified with probability 0.999999 to class 14, which was described above as class of *governmental/public-authority*. The numbers in the column show the estimated frequencies of the subject fillers.

Figure 25 shows the subjects of the intransitive verb *steigen* 'rise' which belong with probability 0.67273 to class 26 which was interpreted above as a class of *gradation/scalar change*.

Similar to the English experiments we observe semantic uniformity

bekanntgeben 14	0.999999	‘make public’
Sprecher	4	‘spokesman’
Polizei	3	‘police’
BundesAmt	3	‘Federal Agency’
BürgerMeister	2	‘mayor’
VorstandsChef	2	‘Chairman of the board’
GeschäftsLeitung	2	‘manager’
Vorstand	2	‘board of management’
unternehmen	1.99996	‘company’
WetterAmt	1	‘meteorological office’
VolksBank	1	‘cooperative bank’

FIGURE 24 Intransitive Lexicon Entry: *bekanntgeben* ‘make public’

steigen 26	0.67273	‘rise’
Zahl	23.333	‘number’
Preis	15.895	‘price’
ArbeitsLosigkeit	10.8788	‘unemployment’
Lohn	9.72965	‘wage’
NachFrage	6.83619	‘demand’
Zins	6.80322	‘interest’
Auflage	5.22654	‘print run’
Beitrag	4.22577	‘contribution’
Produktion	4.21641	‘output’
GrundstuecksPreis	4	‘price of a piece of land’

FIGURE 25 Intransitive Lexicon Entry: *steigen* ‘rise’

in the verbs of scalar change. Figure 26 shows 10 intransitive verbs which take class 14 of a 50-classes model (corresponding to class 26 of the 35-class model) as the most probable class to label their respective subject slots. On the basis the most probable class labels these verbs can be summarised as scalar motion verbs. When compared to linguistic classifications of verbs given by (Schuhmacher 1986), we found an agreement of our classification with the class of “einfache Änderungsverben” (simple verbs of change) except for the verbs *anwachsen* ‘increase’ and *stagnieren* ‘stagnate’ which were not classified there at all.

An example of the two most probable subject-object class pairs of a transitive verb, *senken* ‘lower’ is shown in Figure 27. Class 14 has been introduced before as *governmental/public authority* and class 26 as *gradation/scalar change*.

Figure 28 shows the transitive verb *dauern* ‘last’ selecting the class-pair $(0, 10)$ with probability 0.957095 as semantic label for its subject and object slots. Class 0 can be interpreted as *project/action-class* and class 10 as class of *time*.

0.741467	ansteigen	‘go up’
0.720221	steigen	‘rise’
0.693922	absinken	‘sink’
0.656021	sinken	‘go down’
0.438486	schrumpfen	‘shrink’
0.375039	zurückgehen	‘decrease’
0.316081	anwachsen	‘increase’
0.215156	stagnieren	‘stagnate’
0.160317	wachsen	‘grow’
0.154633	hinzukommen	‘be added’

FIGURE 26 Intransitive Scalar Change Verbs

senken (14, 26) ‘lower’	0.450352
BundesBank - LeitZins ‘Federal bank’ - ‘base rate’	5.81457
BundesBank - Zins ‘Federal bank’ - ‘interest’	2.97838
superMarkt - Preis ‘super market’ - ‘price’	1
SommerGeschäft - Verlust ‘summer business’ - ‘loss’	1
BundesBank - DiskontSatz ‘Federal bank’ - ‘minimum lending rate’	0.99999
senken (14, 14) ‘lower’	0.147857
BundesBank - Lombardsatz ‘Federal bank’ - ‘rate on loanes on security’	0.999973
StrafAndrohung - AbtreibungsQuote ‘threat of punishment’ - ‘abortion rate’	0.96842
StrafAndrohung - AbtreibungsZahl ‘threat of punishment’ - ‘number of abortions’	0.96842
FachHandel - LagerKost ‘stores’ - ‘storage charges’	0.878333
Harmonisierung - sozialNiveau ‘harmonisation’ - ‘social level’	0.764319

FIGURE 27 Transitive Lexicon Entries: *senken* ‘lower’

dauern (0, 10)	0.957095	‘last’/‘go on’
Entwirrung - Zeit	2	‘disentanglement’ - ‘time’
BuergerFrageStunde - Stunde	2	‘question time’ - ‘hour’
Prozess - Jahr	2	‘trail’ - ‘year’
schreckensZeit - Jahr	1	‘scaring time’ - ‘year’
ratenZahlung - Jahr	1	‘buy in installments’ - ‘year’

FIGURE 28 Transitive Lexicon Entry: *dauern* ‘last’

Linguistic Interpretation

In some linguistic accounts, multi-place verbs are decomposed into representations involving (at least) one predicate or relation per argument. For instance, the transitive causative/inchoative verb *increase* is composed of an actor/causative verb combining with a one-place predicate in the structure on the left in Figure 29. Linguistically, such representations are motivated by argument alternations (diathesis), case linking and deep word order, language acquisition, scope ambiguity, by the desire to represent aspects of lexical meaning, and by the fact that in some languages the postulated decomposed representations are overt, with each primitive predicate corresponding to a morpheme. For references and recent discussion of this kind of theory see (Hale and Keyser 1993) and (Kural 1996).

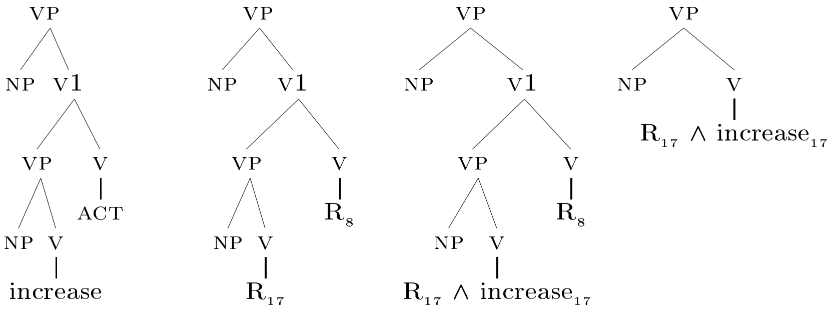


FIGURE 29 First Tree: Linguistic Lexical Entry for Transitive Verb *increase*. Second Tree: Corresponding Lexical Entry with Induced Classes as Relational Constants. Third Tree: Indexed Open Class Root added as Conjunct in Transitive Scalar Motion *increase*. Fourth Tree: Induced Entry for Related Intransitive *increase*.

We will sketch an understanding of the lexical representations induced by latent-class labelling in terms of the linguistic theories mentioned above, aiming at an interpretation which combines computational learnability, linguistic motivation, and denotational-semantic adequacy. The basic idea is that latent classes are computational models of the atomic relation symbols occurring in lexical-semantic representations. As a first implementation, consider replacing the relation symbols in the first tree in Figure 29 with relation symbols derived from the latent class labelling. In the second tree in Fig 29, R_{17} and R_8 are relation symbols with indices derived from the labelling procedure of section 12.6. Such representations can be semantically interpreted in standard ways, for instance by interpreting relation symbols as denoting relations between events and individuals.

Such representations are semantically inadequate for reasons given in philosophical critiques of decomposed linguistic representations; see (Fodor 1998) for recent discussion. A lexicon estimated in the above way has as many primitive relations as there are latent classes. We guess there should be a few hundred classes in an approximately complete lexicon (which would have to be estimated from a corpus of hundreds of millions of words or more). Fodor's arguments, which are based on the very limited degree of genuine interdefinability of lexical items and on Putnam's arguments for contextual determination of lexical meaning, indicate that the number of basic concepts has the order of magnitude of the lexicon itself. More concretely, a lexicon constructed along the above principles would identify verbs which are labelled with the same latent classes; for instance it might identify the representations of *grab* and *touch*.

For these reasons, a semantically adequate lexicon must include additional relational constants. We meet this requirement in a simple way, by including as a conjunct a unique constant derived from the open-class root, as in the third tree in Figure 29. We introduce indexing of the open class root (copied from the class index) in order that homophony of open class roots not result in common conjuncts in semantic representations—for instance, we don't want the two senses of *decline* exemplified in *decline the proposal* and *decline five percent* to have a common entailment represented by a common conjunct. This indexing method works as long as the labelling process produces different latent class labels for the different senses.

The last tree in Figure 29 is the learned representation for the scalar motion sense of the intransitive verb *increase*. In our approach, learning the argument alternation (diathesis) relating the transitive *increase* (in its scalar motion sense) to the intransitive *increase* (in its scalar motion sense) amounts to learning representations with a common component $R_{17} \wedge \mathrm{increase}_{17}$. In this case, this is achieved.

12.6.4 Further Applications

Probabilistic clustering methods for natural language applications mainly focus on the following two tasks: (i) induction of smooth probability models on language data, and (ii) automatic discovery of class-structure in natural language. In the above described application of clustering to lexicon induction we focussed our attention on the second task. There we were interested in the structure of the induced clusters as a statistical semantics underlying the data in question. In other applications the class-structure itself is not of interest, rather data clusters are consulted as general back-up sources of information when information about specific events is sparse or missing in the input. Here smooth clustering models can be used to solve sparse data problems in various application areas. For applications of EM-based clustering in lexical disambiguation in machine translation see (Prescher et al. 2000), or in head-word lexicalisation of probabilistic grammars see (Johnson and Riezler 2000, Riezler et al. 2000).

12.7 Conclusion

In the preceding sections, we presented a framework for the development and training of statistical grammar models and successfully applied it to the acquisition of lexicon information. In particular, we described methods for the extraction of subcategorisation frames for verbs and for the determination of selectional restrictions. The resulting information is easy to use for lexicographers. Our approach has already been applied to German, English, Portuguese and Chinese and will be applied to Greek and Spanish in the near future. In addition, the linguistic information gained in our experiments is valuable for natural-language applications like lexicography, parsing, information retrieval, or machine translation.

In an extensive experiment, we applied semantic clustering techniques to predicate-argument pairs in order to induce semantic classes representing typical predicate-argument relationships. Such classes are not only interesting from a linguistic point of view, but can also be directly used to solve sparse-data problems in natural language modelling.

The mathematically well-defined Expectation-Maximisation algorithm for unsupervised learning was used in all our experiments. Although there is no guarantee that the maximisation of the likelihood of the training data which the EM algorithm performs, also improves the linguistic correctness of the resulting syntactic analyses, our experiments show that in practice this is the case. Gaining more insight into the relationship between linguistic plausibility and likelihood of linguistic analyses will be an interesting future research topic.

References

Abney, Steven. 1996. Chunk Stylebook. Technical report. Seminar für Sprachwissenschaft, Universität Tübingen.

Baker, J. 1979. Trainable Grammars for Speech Recognition. In *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, ed. D. Klatt and J. Wolf, 547–550.

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

Baum, Leonard E., Ted Petrie, George Soules, and Norman Weiss. 1970. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics* 41(1):164–171.

Beil, Franz, Glenn Carroll, Detlef Prescher, Stefan Riezler, and Mats Rooth. 1998. Inside-Outside Estimation of a Lexicalized PCFG for German. –Gold–. In *Inducing Lexicons with the EM Algorithm.* AIMS Report 4(3). Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Beil, Franz, Glenn Carroll, Detlef Prescher, Stefan Riezler, and Mats Rooth. 1999. Inside-Outside Estimation of a Lexicalized PCFG for German. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99).* College Park, MD.

Brown, Peter, Peter deSouza, Robert Mercer, Vincent Della Pietra, and Jenifer Lai. 1992. Class-Based n-gram Models of Natural Language. *Computational Linguistics* 18(4):467–479.

Carroll, Glenn. 1995. *Learning Probabilistic Grammars for Language Modeling.* Doctoral dissertation, Department of Computer Science, Brown University.

Carroll, Glenn. 1997. *Manual pages for* `charge`, `hyparCharge`. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Carroll, Glenn, and Mats Rooth. 1998. Valence Induction with a Head-Lexicalized PCFG. In *Proceedings of EMNLP-3.* Granada.

Charniak, Eugene. 1996. Tree-Bank Grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI '96)*, 1031–1036.

Dagan, Ido, Lillian Lee, and Fernando Pereira. 1999. Similarity-Based Models of Word Cooccurrence Probabilities. In *Machine Learning*, 43–69. Special issue on natural language learning.

de Lima, Erika F. 2001. The Automatic Acquisition of Lexical Information from Portugese Text Corpora with a Probabilistic Context-Free Grammar. Unpublished Manuscript, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the *EM* Algorithm. *Journal of the Royal Statistical Society* 39(B):1–38.

Fodor, Jerry A. 1998. *Concepts: Where Cognitive Science Went Wrong.* Oxford: Oxford Cognitive Science Series.

Hale, K., and S.J. Keyser. 1993. Argument Structure and the Lexical Expression of Syntactic Relations. In *The View from Building 20*, ed. K. Hale and S.J. Keyser. Cambridge, MA: MIT Press.

Hindle, Donald, and Mats Rooth. 1993. Structural Ambiguity and Lexical Relations. *Computational Linguistics* 19:103–120.

Hockenmaier, Julia. 1999. Parsing Unsegmented Chinese Text with a Head-Lexicalised PCFG. Master's thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Hofmann, Thomas, and Jan Puzicha. 1998. Unsupervised Learning from Dyadic Data. Technical Report TR-98-042. Berkeley, CA: International Computer Science Insitute.

Johnson, Mark, and Stefan Riezler. 2000. Exploiting Auxiliary Distributions in Stochastic Unification-Based Grammars. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000).* Seattle, WA.

Kural, Murat. 1996. *Verb Incorporation and Elementary Predicates.* Doctoral dissertation, University of California, Los Angeles.

Lapata, Maria, Frank Keller, and Sabine Schulte im Walde. To appear. Verb Frame Frequency as a Predictor of Verb Bias. *Journal of Psycholinguistic Research.*

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

Levin, Beth. 1993. *English Verb Classes and Alternations. A Preliminary Investigation.* Chicago/London: The University of Chicago Press.

McLachlan, Geoffrey J., and Thriyambakam Krishnan. 1997. *The EM Algorithm and Extensions.* New York: Wiley.

Ney, Hermann, Ute Essen, and Reinhard Kneser. 1994. On Structuring Probabilistic Dependencies in Stochastic Language Modelling. *Computer Speech and Language* 8:1–38.

Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional Clustering of English Words. In *Proceedings of the 31th Annual Meeting of the Association for Computational Linguistics (ACL'93).* Columbus, Ohio.

Prescher, Detlef, Stefan Riezler, and Mats Rooth. 2000. Using a Probabilistic Class-Based Lexicon for Lexical Ambiguity Resolution. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, 649–655. Saarbrücken, Germany.

Riezler, Stefan, Detlef Prescher, Jonas Kuhn, and Mark Johnson. 2000. Lexicalized Stochastic Modeling of Constraint-Based Grammars using Log-Linear Measures and EM Training. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00).* Hong Kong.

Rooth, Mats. 1995. Two-Dimensional Clusters in Grammatical Relations. In *Symposium on Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity.* AAAI'95 Spring Symposium Series. Stanford University.

Rooth, Mats. 1998. Two-Dimensional Clusters in Grammatical Relations. In *Inducing Lexicons with the EM Algorithm.* AIMS Report 4(3). Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Rooth, Mats, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1998. EM-Based Clustering for NLP Applications. In *Inducing Lexicons with the EM Algorithm.* AIMS Report 4(3). Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Rooth, Mats, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a Semantically Annotated Lexicon via EM-Based Clustering. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99).* Maryland.

Saul, Lawrence K., and Fernando Pereira. 1997. Aggregate and Mixed-Order Markov Models for Statistical Language Processing. In *Proceedings of EMNLP-2.*

Schiller, Anne, and Chris Stöckert. 1995. *DMOR.* Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Schmid, Helmut. 1999. *YAP: Parsing and Disambiguation with Feature-Based Grammars.* Doctoral dissertation, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Schmid, Helmut. 2000. *LoPar: Design and Implementation.* Arbeitspapiere des Sonderforschungsbereiches 340, No. 149. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Schmid, Helmut, and Sabine Schulte im Walde. 2000. Robust German Noun Chunking With a Probabilistic Context-Free Grammar. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, 726–732. Saarbrücken, Germany.

Schuhmacher, Helmut. 1986. *Verben in Feldern. Valenzwörterbuch zur Syntax und Semantik deutscher Verben.* Berlin: De Gruyter.

Schulte im Walde, Sabine. 1998. Automatic Semantic Classification of Verbs According to their Alternation Behaviour. Master's thesis, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart.

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

Schulte im Walde, Sabine. 2000b. *The German Statistical Grammar Model: Development, Training and Linguistic Exploitation.* Arbeitspapiere des Sonderforschungsbereiches 340, No. 162. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Schulze, Bruno Maximilian. 1996. *GermLem – ein Lemmatisierer für deutsche Textcorpora.* Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.