Valence Induction with a Head-Lexicalized PCFG

Glenn Carroll, Mats Rooth, Detlef Prescher, and Marc Light
IMS, University of Stuttgart

May 3, 1998

1 Introduction

This chapter explains the theory underlying the lexicon learning system which was implemented at the University of Stuttgart, and reviews the experiment with the induction of a German lexicon which were performed in the Sparkle project.[1]

The problems which motivate our work relate to the complexity and size of the lexicon; to the massive ambiguity in grammatical analysis which is observed in naturally occurring language, particularly written language; and to the scale of the texts which we would like to analyze computationally. We believe that in order to address these problems, computational linguistic technologies should be based on a combination of symbolic and numerical modeling, should embody approximations together with exact theories, and should constructed in a way which makes it possible to learn much of the information which is deployed from large samples of language use. In the research described in this chapter, probability theory and probabilistic algorithms are used to establish a link between symbolic and numerical models, to deal with ambiguity, and to allow models of the lexicon to be learned from text data.

The transformation of a symbolic model into a numerical one can be illustrated by the representation of the subcategorization (valence) of the verb *question* in a lexicon for English. An ordinary symbolic lexicon directly or indirectly specifies the set of valence frames with which *question* can combine to form a verb phrase. Five of the possibilities are:

(i) a noun phrase (*ask* · *a question*)

(ii) a for-prepositional phrase (*ask* · *for help*)

[1]In addition to the authors, Chris Stöckert, Veerle van Geenhoven, Sabine Schulte im Walde, and Karin Müller worked on various aspects of the experiment.

(iii) a noun phrase followed by a to-infinitive (*ask* · *him* · *to leave*)

(iv) an indirect question (*ask* · *who ate the cookies*)

(v) a base form *that* complement (*ask* · *that she pay pay her phone bill*)

In a probabilistic lexicon, this set is replaced by a probability distribution over possible valences. Such a distribution can be represented as a table which specifies a number $p[f]$ in the interval $[0, 1]$ for each verbal frame f, such that the entries in the table add up to 1. For instance, the table might specify the values $p[\textsc{np}] = 0.3$ and $p[\textsc{np} \cdot \textsc{vto}] = 0.1$. The fact that such a table describes the relative frequency of possible frames turns out to be a key to the problems of ambiguity, robustness, and learnability.

The remainder of this chapter is organized as follows. Section 2 describes a lexicalized probabilistic grammar formalism known as head-lexicalized probabilistic context free grammar, and the algorithm by means of which the probability parameters of such a grammar are learned from data. Section 3 outlines the grammar and text corpus which were employed in the experiment on German. In section 4, we explain how the software systems documented in reports 3.2 (parsing) and 7.1 (lexicon induction) were deployed in the experiment, and evaluate results with reference to a lexicon induction test suite. Section 5 discusses future work.

2 A lexicalized probabilistic grammar formalism

A probabilistic grammar can be used to assign numeric weights are assigned to tree analyses. Our method for lexicalizing a probabilistic grammar is based on the idea of projecting lexical heads up the tree, so that they are accessible at the levels where they should condition various probabilistic choices. Figure 1 is a tree analysis of the verb phrase *has asked a perfectly reasonable question* in an X-bar grammar of of English described in Carroll and Rooth (ms). Together with grammatical categories such as NP (noun phrase) and N1 (N bar), each node is labeled with a lexical head. For instance, the lexical label for the entire finite verb phrase (VFP) is *asked*, and the lexical label for the object NP *a perfectly reasonable question* is *question*. Projected heads are used to condition two kinds of probabilistic choices:

- Complementation and other phrase structure rules. The choice or the expansion VFC · NP for ⟨VFC, question⟩ is conditioned on the the head verb *question* as well as the parent category VFC. This allows the probabilistic weighting of tree structure to reflect the fact that *ask* frequently takes a transitive frame.

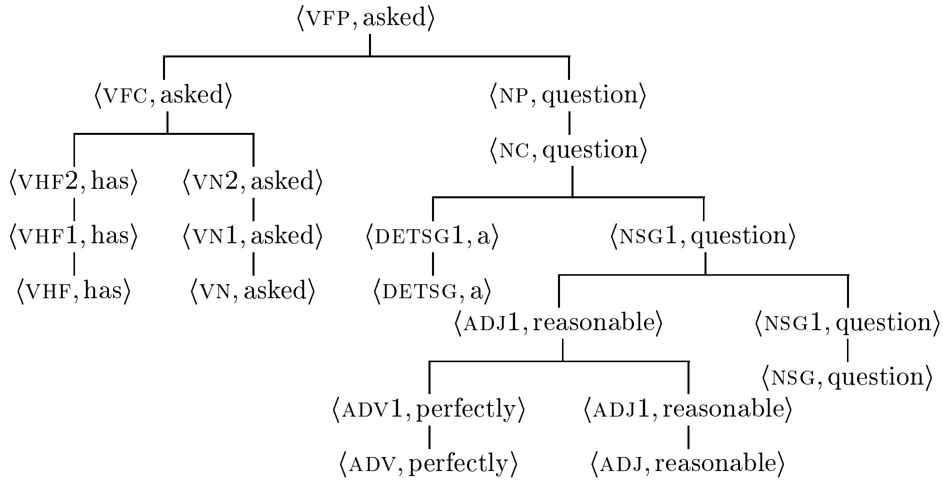

Figure 1: Projection of heads.

- Lexical content. Head words are chosen at the top of the chain of heads, where they occur as non-head daughters. In the example, *question* is chosen at the level of the VFC expansion, and is conditioned on ASKED, the head of the parent category. This allows the probabilistic weighting of tree structure to reflect the fact that *ask question* is a verb/object collocate.

Event polynomials

Probabilistic weights for tree analyses, sentences, and text corpora are defined in terms of polynomials in a large set of variables; the variables are known as events. Consider the tree analysis in figure 2 for the N bar *big big problem*.

The head relation is used to transform the phrase structure tree into an *event tree* labeled with certain rule and lexical events, as indicated in Figure 3. The rule events are written in square brackets, and have the following interpretations:

(1)	[A1,big → A']	A bar headed by *big* expanded as A
	[N1,problem → N']	N bar headed by *problem* expanded as N
	[N1,problem → A1 N1']	N bar headed by *problem* expanded as A1 N1

The lexical events are written in round brackets, and have the following interpretations:

(2)	(N1,problem,A1,big)	*big* chosen as the head of A1 child of N1 headed by problem

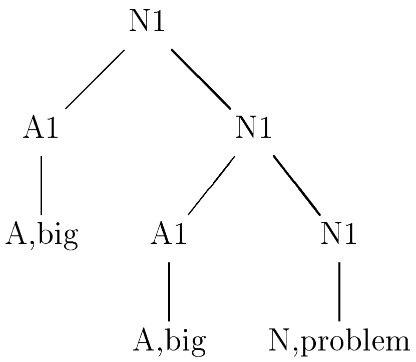

Figure 2: A tree analysis.

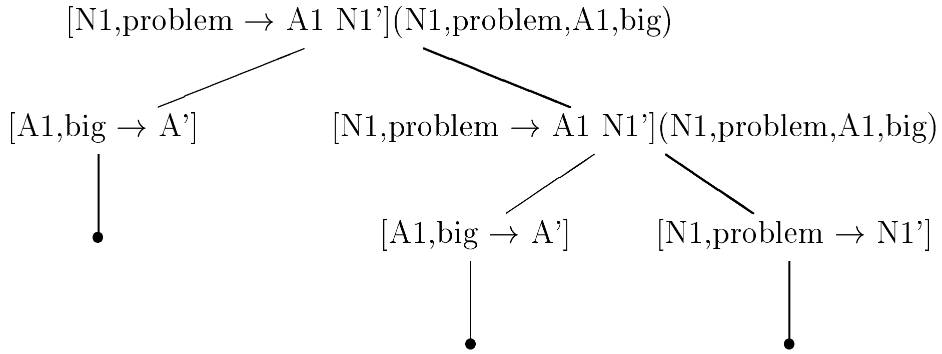

Figure 3: The corresponding event tree.

The *event monomial* $e(\tau)$ for an event tree τ is the symbolic product of the events labeling the tree. The event monomial corresponding to the tree in Figure 3 is given in Figure 4.

Where E is the set of rule and lexical parameters for a given grammar and lexicon, a lexicalized probabilistic grammer is identified with a function:

$$p : E \to [0, 1]$$

which meets the following conditions:

(i) Fixing any lexicalized non-terminal $\bar{n}\bar{w}$,

$$\sum_{\alpha} p([n, w \to \alpha]) = 1$$

(ii) Fixing any lexicalized non-terminal $\bar{n}\bar{w}$ and possible non-head daughter category x, $\sum_w p((x, \bar{n}, \bar{w}, w)) = 1$.

$$[\text{A1,big} \rightarrow \text{A'}]^2\ [\text{N1,problem} \rightarrow \text{N'}]^1\ [\text{N1,problem} \rightarrow \text{A1 N1'}]^2$$
$$(\text{N1,problem,A1,big})^2(\textsc{startc},\textsc{startw},\text{N1,problem})^1$$

Figure 4: The corresponding event monomial.

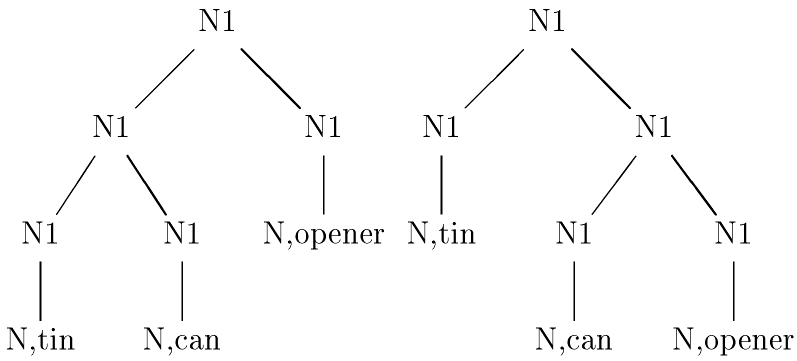

Figure 5: Analyses for *tin can opener*.

The probabilistic weight of a tree is obtained by evaluating its event monomial. Given a parameter setting p, the probabilistic weight $w(\tau, p)$ is defined as the value of the event monomial $e(\tau)$ at p. Using subscripting to notate evaluation of monomials and polynomials, we have:

$$w(\tau, p) \stackrel{\text{def}}{=} [e(\tau)]_p$$

Polynomials and weights for sentences and corpora

The syntactic structure of phrases and sentences is often ambiguous; for instance, the string *tin can opener* has two analyses as a noun compound given in Figure 5.

Let us assume a grammar where N1 is the start symbol, and where the string $\sigma =$ *tin can opener* has exactly the tree analyses above. The event polynomial $e(\sigma)$ for σ is the polynomial sum of the event monomials for the tree analyses of σ. In the example, the event polynomial is the one in Figure 6.

A corpus $\mathcal{C}$ is a finite sequence of sentences. The event polynomial $e(\mathcal{C})$ for $\mathcal{C}$ is defined to be the polynomial product of the event polynomials for the sentences in $\mathcal{C}$. Probabilistic weights for sentences and corpora are obtained by evaluating their event polynomials:

$$w(\tau, \mathcal{C}) \stackrel{\text{def}}{=} [e(\mathcal{C})]_p$$

$$[\text{N1,tin} \rightarrow \text{N'}]^1\ [\text{N1,can} \rightarrow \text{N'}]^1\ [\text{N1,opener} \rightarrow \text{N'}]^1\ [\text{N1,opener} \rightarrow \text{N1 N1'}]^1$$
$$[\text{N1,can} \rightarrow \text{N1 N1'}]^1 (\text{N1,can,N1,tin})^1 (\text{N1,opener,N1,can})^1 (\textsc{startc},\textsc{startw},\text{N1,opener})^1 +$$
$$[\text{N1,tin} \rightarrow \text{N'}]^1\ [\text{N1,can} \rightarrow \text{N'}]^1\ [\text{N1,opener} \rightarrow \text{N'}]^1\ [\text{N1,opener} \rightarrow \text{N1 N1'}]^2$$
$$(\text{N1,opener,N1,tin})^1 (\text{N1,opener,N1,can})^1 (\textsc{startc},\textsc{startw},\text{N1,opener})^1$$

Figure 6: The corresponding event polynomial.

In line with the framework of Baum and Sell (1968), the problem estimating probability parameters from a corpus $\mathcal{C}$ (i.e. learning a probabilistic lexicon) is now a matter of choosing parameters p which maximize the value of the polynomial $[e(\mathcal{C})]$, subject to the constraints on p defined above. In statistical terms, this is the method of maximum likelihood estimation. The problem of finding global maxima for the the polynimials arising from context free grammars and lexicalized context free grammars is intractable. The algorithm of Baum and Sell produces local improvements in the probability of the corpus, and could in principle be used to find a local maximum of the corpus polynomial. However, in our models even this procedure would be too time consuming, perhaps involving years of computation. We instead use a modified algorithm which iteratively increases the value of the corpus polynomial, without maximizing it.

In our interpretation, the function p is a probabilistic lexicon. A good lexicon for English should assign a fairly high value to the rule parameter [VP,ask → V' NP'], because *ask* is used fairly frequently with a transitive valence frame. And it should assign a fairly high value to the lexical parameter (VP,ask,NP,question), because *question* is a frequent object collocate of *ask*. Experiments suggest that these specifically linguistic requirements in fact end up being satisfied in a probabilistic lexicon which is trained by an algorithm which seeks to increase the value of the corpus polynomial (Carroll and Rooth ms). We now turn to this algorithm.

2.1 Parameter estimation

The algorithm of Baum and Sell (1968) for constrained maximization of polynomials involves computing an event count determined by a given polynomial ϕ and parameter setting p; in our application, ϕ is the corpus event polynomial $e(\mathcal{C})$. The implemented algorithm works sentence by sentence, and we will define $c(\sigma, p)$ for a sentence σ directly in terms of the event monomials for trees τ with yield σ. In the equations below, $e(\tau)$ and $e(\sigma)$ are viewed as a vectors of integers indexed by $E(G)$. The sum in the first equation is a pointwise sum of vectors indexed by $E(G)$; juxtaposition of the fraction with $e(\tau)$ represents a scalar product, yielding a vector of reals.

(3) $\quad c(\sigma, p) \stackrel{\text{def}}{=} \sum_{\tau|y(\tau)=\sigma} \frac{[e(\tau)]_p}{[e(\sigma)]_p} e(\tau)$

(4) $\quad [e(\sigma)]_p = \sum_{\tau|y(\tau)=\sigma} [e(\tau)]_p$

The second equation can be used (though in this form not efficiently) to calculate the probability of a sentence σ; it is a consequence of the fact that evaluation of polynomials is a homomorphism from the ring of polynomials to the reals (e.g. S. Lang 1993).

Where z is an event, $c(\sigma, p)(z)$ has the probabilistic interpretation of the expected number of occurrences of the the event z in a tree with yield σ. The sample space consists of labeled trees, with probability measure parameterized by p, as defined above for finite trees; the random variable for which a conditional expectation is computed maps a tree τ to $e(\tau)(z)$. The conditioning event for the conditional expectation is that $y(\tau) = \sigma$.

Algorithmically, we compute the event count for a sentence σ with the inside-outside algorithm in a parse forest representation of the set of tree analyses of σ. A parse forest is an and-or graph representing a set of tree analyses (B. Lang 1989), in which event counts can be computed efficiently using a graph variant of the inside-outside algorithm.

Given a parameter setting p, event counts are computed and summed over the sentences in the corpus. Using a suitable definition of the event-count determined by a polynomial and a parameter vector, this can be shown to compute the event-count for the corpus. In the algorithm of Baum and Sell (1968) new parameter values would be defined as relative frequencies of event counts, equivalent to maximum-likelihood estimation based on estimated frequencies of hidden events (Dempster, Laird, and Rubin 1977). This general algorithm is known as the EM algorithm. Our procedure differs in two ways. First, we use smoothing scheme in order to deal with the size of the parameter space and the resulting problems that (i) counts are zero for the majority of events, and (ii) the parameter space is too large to be represented directly in computer memory. The smoothing procedure has the effect that events which are observed with a low or null frequency are assigned default, unlexicalized probabilities. Smoothing has the consequence any valence frame can combine with any verb, though perhaps with very low probability. Definitions connected with smoothing are given in Carroll and Rooth (ms).

Second, computation time required for parsing and for computing and summing estimated frequencies is such that it is not feasible to estimate a model by parsing the whole corpus to estimate event frequencies, estimating a new probability model, and repeating the whole procedure iteratively until convergence. Instead, probability parameters are reset at intervals as the corpus is parsed. First a non-lexicalized model is trained and used to collect frequencies for lexicalized events on about a million words of text. A lexicalized probability model is estimated from these frequencies. Then a further segment of the corpus is parsed with the lexicalized model to estimate event freqencies in that segment. These frequecies are merged with the earlier event freqencies, a new lexicalized model is estimated, and the procedure iterates. Various training schemes are possible in which frequency counts which come from earlier parts of the corpus (and thus were parsed with a less well-trained lexicalized model, or an unlexicalized model) are eventually dropped or scaled.

3 The German grammar

The learning procedure described above is implemented in the lexicon acquisition device delivered in WP7.1. The method is generic, and we believe it can be applied to any natural language for which a large text corpus (10-100 million words), morphology, and linguistic expertise are available. The learning procedure has previously been applied in an experiment on English where 9/10 of the British National Corpus was mapped to a probabilistic lexicon (Carroll and Rooth ms). This English model may be used as a target probabilistic language model in one of the Sparkle multilingual demonstrators. The main focus of our work in Sparkle, however, is on creating a probabilistic lexicon for German, together with methodologies for tuning the lexicon to specific vocabularies and domains. The lexicon and tuning methodologies are to be validated in the Spoken Dialogue demonstrator (WP11) constructed at Daimler Benz Research Institute.

The parsing and acquisition system is based on context free grammar, rather than a formalism embodying features. To the extent that feature agreement is to be imposed, features have to be encoded in the symbols of the context free grammar. For instance, the German grammar uses the non-terminal symbol

ADJA1.Dat.Neut.Sg.Sw

with the interpretation of a dative, neuter, singular, weak A bar. This symbol is atomic from the point of view of the parser. It is not feasible to develop grammars at this level of specificity; underlying grammars should be expressed more abstractly. For this purpose, we employed the Helmut Schmid's YAP system (Schmid 1997). Rules in this formalism are headed context rules annotated with typed feature structures. The system includes a parser in which a context free parse forest is constructed and supplied with feature annotations, is usable with a large lexicon (tens of thousands of words), and includes a simple development and debugging environment. Below is a rule written in the YAP formalism which expands an noun chunk (NC) as an article followed by an N bar.

(5)
```
NC {Pers=thd; Num=q; Gender=g; Case=c;}   ->
          ART1 {Num=q; Case=c; Gender=g; Stre=s; }
          'NN1   {Num=q; Case=c; Gender=g; Stre=s; };
```

The YAP system includes a facility for compiling the feature grammar into a headed context free grammar by incorporating features in grammar symbols; this is how the grammar used by the acquisition device is generated. The feature set used in the source grammar is fairly complete. For instance, the type declaration for the category NC1 (noun phrase) includes four features.

(6)
```
category NC1 {
  PERS   Pers;
  NUM    Num;
  GENDER Gender;
  CASE   Case;
 };
```

In practice, in order to control the size of the generated grammar, not all features were incorporated. In the example above, only the case feature was included at the noun phrase level in the grammar used by the probabilistic parser. This has the consequence that agreement between subject and tensed verb is not enforced, although person, number, and gender agreement are enforced within the noun phrase.

Compilation increases the number of categories by about a factor of 4; the source typed feature grammar uses 219 categories, while the compiled grammar uses 840 categories. The blowup in the number of rules is more severe; 31166 headed context free rules are generated from 1396 source rules.

Feature encoding of valences

The primary goal of the experiment was induction of valences for individual verbs. For instance, we would like to learn that the verb *zustimmen* (agree to/approve) combines with nominative and dative complements:

(7) $[_{\text{nom}}$ der Magistrat] hat $[_{\text{dat}}$ den Plänen] zugestimmt.
The municipal authorities have approved the plans.

Because of the grammatical phenomena of topicalization, scrambling, and verb placement, this nominative/dative frame can be expressed in a number of surface word orders. In order to allow frequencies to be pooled among these different word orders, valence frames were encoded as a feature at an ad-hoc node above the clause level. The rule which introduces a nominative/dative frame into and active verb second clause is

V2C15 $\rightarrow$ V2C1.Nd.Act'

where Nd is an encoding of the nominative/dative valence. Further down in the tree, the complements may be realized in various orders. Instances of *zustimmen* taking the the nominative/dative frame are reflected in frequency counts for the lexicalized rule V2C15,zustimmen $\rightarrow$ V2C1.Nd.Act'.

The feature grammar rules which introduce various word orders for complements and verbs are generated by a Perl script which puts complements in various orders. This permutation procedure produces 1029 complementation rules from 339 source rule descriptions. These verbal complementation rules account for the bulk of the 1396 rules in the feature grammar.

The state grammar

The grammar used by the probabilistic parser includes additional *state rules* which allow clause and phrase units to be strung together without constructing a conventional grammatical analysis. This is a robustness feature which increases the fraction of sentences in the corpus which can be parsed. State rules are of the form

$$\mathrm{X{:}Y} \quad \rightarrow \quad \mathrm{Y}' \quad \mathrm{Y{:}Z}$$

where x, y, and z are clausal or phrasal categories. The rule represents the conditioning of one phrasal/clausal category on the previous two such categories, as in a second order Markov model.

Morphology and text corpus

As used in experiments at the University of Stuttgart, the parser makes direct calls to a two-level morpology (Schiller ms); the delivered system uses is full-form word list. The corpus for the experiment was the Frankfurter Rundschau corpus, comprising approximately 40 million tokens. The corpus is represented in the CQP/XKWIC corpus query system (Christ ms), making it easy to conduct experimements with specific subparts of the corpus, such as those containing a given verb lemma.

4 Lexicon induction experiment and evaluation

This section reports on an experiment in which a general probabilistic lexicon was trained on 2 million words of text, and then used in lexicalized parsing to compute estimated valence frequencies for all the occurrencees in the FR corpus of 30 selected verbs. This allowed an evaluation of valence induction to be based on a sufficient number of verb occurrences, while keeping training times within bounds established by project deadlines. Thus the lexicon learned in the experiment consists of (i) a general probabilistic lexicon trained on 2 million words of text, and (ii) for each of 30 verb lemmas, a probabilistic lexicon trained by means of lexicalized parsing using the first lexicon on all of the occurrences of that verb in the FR corpus.

To illustrate the information learned, consider the verb *zustimmen*, which predominately takes a nominate/dative valence frame. In the FR, this verb

freq	prob	rule	
113.1	0.6278	V2C15 → V2C1.Nd.Act'	(nominative/dative)
22.84	0.1268	V2C15 → V2C1.N.Act'	(nominative)
10.75	0.05965	V2C15 → V2C1.Na.Act'	(accusative)
5.54	0.03075	V2C15 → V2C1.N_sub.Act'	(sbar complement)

Figure 7: Estimated valence frequencies for *zustimmen*.

zustimmen Dat		verursachen Acc	
noun	freq	noun	freq
Entwurf (design)	10.4941	Unfall (accident)	42.8253
Verkauf (sale)	8.53122	Kosten (costs)	33.6943
Vertrag	7.45123	Schaden	27.7507
Antrag	7.11451	Schäden	10.992
Plan	6.98768	Ärger	8.27871
Kompromiß	6.6572	Verkehrsunfall	6.47082
Änderung	6.39134	Brand	5.56298
Verkehrskonzept	5	Absturz	4.94972
Konzept	4.97326	Lärm	4.17167
Projekt	4.65944	Feuer	4.03284

Figure 8: Collocate frequencies

occurs as the head of a verb second clause with a frequency which is estimated by the acquisition system to be 180.169. This frequency is divided among a variety of valence frames. The ones with highest estimated frequency are given in Figure 7.

A second kind of learned information characterizes word collocates. Figure 8 gives the most frequent dative object collocates of *zustimmen* (agree to) and the most frequent accusative object collocates for *verursachen* (cause). The collocates are intuitively plausible.

These data are drawn form the lexicons learned for specific verbs, but because these subcorpora contain all occurrences of a given verb, they are equivalent to what would be obtained from entire FR corpus, modulo differences in training schedule. They indicate the richness and degree of specificity of the collocation information which would be available for each word (or each frequent word) in a lexicon trained on the whole FR corpus.

Evaluation

The learned valence distributions were evaluated against a lexicon which has been semi-automatically acquired from corpora at the University of Stuttgart by Judith Eckle-Kohler based on the exploitation of unambiguous surface patterns of predicate argument structures, which constitute only a part of the overall corpus data (Eckle-Kohler 1998; Kuhn, Eckle-Kohler, and Rohrer 1998). Seven common frames were used for the preliminary evaluation described here. In order to evaluate the probabilistic dictionary entry for a given verb, the probability distribution over frames must be mapped to a set of frames. This was done by establishing a cutoff probability value.

Different cutoffs were used for different frames, and indeed for different verbs. The 30 verb set was used for both training and testing data, using the technique of cross-validation. Fixing a frame f, for each verb v an optimal cutoff point was set using the other 29 verbs as training data. The resulting cutoff was used to map the floating point probability for the verb/frame combination $\langle v, f\rangle$ to Boolean (yes/no) value $L(v, f)$. The procedure produces a learned lexicon L of the ordinary kind, namely a relation between verbs and frames. Cross-validation can be expected to give results similar to what would be obtained on a 30 verb test set, setting a single cutoff value for each frame on a separate 29 word training set.

Precision and recall figures were computed for each frame. For a given frame f, precision is defined to the proportion of entries $\langle v, f\rangle$ in the learned dictionary such that $\langle v, f\rangle$ is also in the standard dictionary. Recall is the fraction of entries $\langle v, f\rangle$ in the standard dictionary such that $\langle v, f\rangle$ is also in the learned dictionary. Both precision and recall take values in the interval [0,1], and higher figures are better.

The cutoffs in the cross-validation procedure were set to the minimal value where precision was greater than equal to recall, as computed on the 29 verb training set. Carroll and Rooth (ms) used a similar but somewhat more sophisticated criterion for setting the cutoffs, and used separate training and testing sets rather than cross-validation.

Results for the seven frames are given in Figure 9. Since the corpus on which the parser was trained is so small (2 million words), these numbers evaluate the extraction of frame frequencies by a poorly lexicalized parser, rather than itererative EM estimation of valence distributions. In particular, the parser could not take advantage of good frame distributions for the verbs being tested, since these verbs have small frequencies in the 2 million word corpus. Results can be expected to improve as the lexicalized parser is trained on larger corpora.

precision	recall	frame
0.5	0.67	Nom-Dative
0.88	0.92	Nom-Acc
0.77	0.64	Nom-Sbar
0.44	0.5	Nom-Acc-Dat
0.54	0.6	Nom-Prep
0.5	0.5	Nom
0.5	0.5	Nom-Infinitive

Figure 9: Precision and recall for seven frames.

5 Discussion

The experiment described above was the first experiment of substantial scale with the German parser. It can be considered a proof of concept for the parsing and lexicon training procedure as applied to German. In the experiment, parsing times were a hindrance (about a week for a run of the experiment), in part because only one workstation with sufficient internal memory (500MB) to store the German probability model was available. Both the size of probability models and parsing times are higher for German than for English.

One planned change in the lexicon should ameliorate these problems. In the model used in the present experiment, the inflected verb forms rather than lemmas were used as the projected heads of phrases. This not only reduces frequencies by splitting them between inflectional forms (resulting in failure to generalize across inflectional forms), but also increases the size of the probability model (perhaps by a factor of 3) because there are different parameters for different inflectional forms. The latest parser implementation includes a facility for working with lemmas rather than inflected words as projected heads. This feature will also be the basis for incorporating semantic classes into the parser. Here classes (such as EVENT or PERSON, or automatically induced counterparts thereof) will serve as projected heads.

We also plan to increase internal memory on laboratory workstations, in order to be able to train models on 5 or more machines working in parallel, as was done in the English experiments. Further work on grammar debugging is also planned, based on the results of probabilistic parsing. Successive probabilistic lexicons will be made available for work in WP11 as they are produced. Collocation parameters learned by the system will be used in work at Stuttgart on semantic type induction.

Bibliography

Steven Abney [1991], "Parsing by Chunks," in *Views on Phrase Structure*, D. Bouchard & K. Leffel, eds., Kluwer Academic Publishers.

Steven Abney [to appearr], "Chunks and dependencies: Bringing processing evidence to bear on syntax," in *Computational Linguistics and the Foundations of Linguistic Theory*, CSLI.

J. K. Baker [1979], "Trainable grammars for speech recognition," *Proceedings of the Spring Conference of the Acoustical Society of America*, Cambridge, MA.

Leonard E. Baum & George R. Sell [1968], "Growth Transformations for Functions on Manifolds," *Pacific Journal of Mathematics* 27.

Taylor L. Booth & Richard A. Thompson [May 1973], "Applying Probability Measures to Abstract Languages," *IEEE Transactions on Computers* C-22, 442–450.

Oliver Christ [1995], "A Modular and Flexible Architecture for an Integrated Corpus Query System," *Manuscript* .

R.O. Duda & P.E. Hart [1973], in *Pattern Classification and Scene Analysis*, Wiley, New York.

Judith Eckle-Kohler [1998], "Methods for quality assurance in semi-automatic lexicon acquisition from corpora," *To appear in Proceedings of EURALEX'98* .

Judith Eckle & Ulrich Heid [1996], "Extracting raw material for a German subcategorization lexicon from newspaper text," *Proceedings of 4th International Conference on Computational Lexicography* , Budapest.

Gerald Gazdar, Ewan Klein, Geoffrey K. Pullum & Ivan A. Sag [1985], *Generalized Phrase Structure Grammar*, Harvard University Press, Cambridge, MA.

Jane Grimshaw [1990], *Argument Structure*, MIT Press, Cambridge, MA.

R. Jackendoff [1977], $\overline{X}$ *syntax: A study in phrase structure.*, MIT Press, Cambridge, MA.

S. M. Katz [1980], "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech and Signal Processing,* 35, 400–401.

Jonas Kuhn, Judith Eckle-Kohler & Christian Rohrer [1998], "Lexicon Acquisition with and for Symbolic NLP-Systems - a Bootstrapping Approach," *To appear in Proceedings of LREC'98* , Granada, Spain.

B. Lang [1989], "A Uniform Formal Framework for Parsing," in *Proceedings of the 1st International Workshop on Parsing Technologies*, Carnegie-Mellon University, Pittsburgh, PA, 28–42.

Serge Lang [1993], in *Algebra, Third edition*, Addison-Wesley, Reading, MA.

Hermann Ney, Ute Essen & Reinhard Kneser [1994], "On structuring probabilistic dependences in stochastic language modelling," *Computer Speech and Language* 8, 1–38.

Carl Pollard & Ivan A. Sag [1987], in *Information-Based Syntax and Semantics, Vol. 1: Fundamentals.*, CSLI.

Anne Schiller [Oktober 1993.], "Deutsche Flexions- und Kompisitionsmorphologie auf 2-Ebenen-Basis," *Manuscript* .

Helmut Schmid [September, 1997], "Parsing by Successive Approximation," *Proceedings of International Workshop on Parsing Technologies*, Cambridge, MA.

Timothy Stowell [1981], "Origins of Phrase Structure.," MIT, Doctoral dissertation.

G. Wahba & S. Wold [1975], "A completely automatic French curve: fitting spline functions by cross validation," *Commun. Statist.* 4, 1–17.