A Novel Disambiguation Method For Unification-Based Grammars Using Probabilistic Context-Free Approximations

Bernd Kiefer, Hans-Ulrich Krieger, Detlef Prescher
{kiefer|krieger|prescher}@dfki.de
Language Technology Lab, DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany

Abstract

We present a novel disambiguation method for unification-based grammars (UBGs). In contrast to other methods, our approach obviates the need for probability models on the UBG side in that it shifts the responsibility to simpler context-free models, indirectly obtained from the UBG. Our approach has three advantages: (i) training can be effectively done in practice, (ii) parsing and disambiguation of context-free readings requires only cubic time, and (iii) involved probability distributions are mathematically clean. In an experiment for a mid-size UBG, we show that our novel approach is feasible. Using unsupervised training, we achieve 88% accuracy on an exact-match task.

1 Introduction

This paper deals with the problem of how to disambiguate the readings of sentences, analyzed by a given unification-based grammar (UBG).

Apparently, there are many different approaches for almost as many different unification-based grammar formalisms on the market that tackle this difficult problem. All approaches have in common that they try to model a probability distribution over the readings of the UBG, which can be used to rank the competing analyses of a given sentence; see, e.g., Briscoe and Carroll (1993), Eisele (1994), Brew (1995), Abney (1997), Goodman (1997), Bod and Kaplan (1998), Johnson et al. (1999), Riezler et al. (2000), Osborne (2000), Bouma et al. (2001), or Schmid (2002).

Unfortunately, most of the proposed probability models are not mathematically clean in that the probabilities of all possible UBG readings do not sum to the value 1, a problem which is discussed intensively by Eisele (1994), Abney (1997), and Schmid (2002).

In addition, many of the newer approaches use log-linear (or exponential) models. Schmid (2002) outlines a serious problem for these models: log-linear models prevent the application of dynamic programming methods for the computation of the most probable parse, if complex features are incorporated. Therefore the run-time complexity of the disambiguation algorithm is linear in the number of parses of a sentence. If the number of parses grows exponentially with the length of the sentence, these approaches are simply impractical.

Our approach obviates the need for such models on the UBG side in that it shifts the responsibility to simpler CF models, indirectly obtained from the UBG. In more detail, the kernel of our novel disambiguation method for UBGs consists of the application of a context-free approximation for a given UBG (Kiefer and Krieger, 2000) and the exploitation of the standard probability model for CFGs.

In contrast to earlier approaches to disambiguation for UBGs, our approach has several advantages. Firstly, probabilistic modeling/training of context-free grammars is theoretically well-understood and can be effectively done in practice, using the inside-outside algorithm (Lari and Young, 1990). Secondly, the Viterbi algorithm enables CFG parsing and disambiguation in cubic time, exploiting dynamic programming techniques to specify the maximum-probability parse of a given sentence. Thirdly, probability distributions over the CFG trees are mathematically clean, if some weak conditions for this desired behaviour are fulfilled (Booth and Thompson, 1973).

In the rest of the paper, we present the context-free approximation, our novel disambiguation approach, and an experiment, showing that the approach is feasible.

2 Context-Free Approximation

In this section, we briefly review a simple and intuitive approximation method for turning unification-based grammars, such as HPSG (Pollard and Sag,

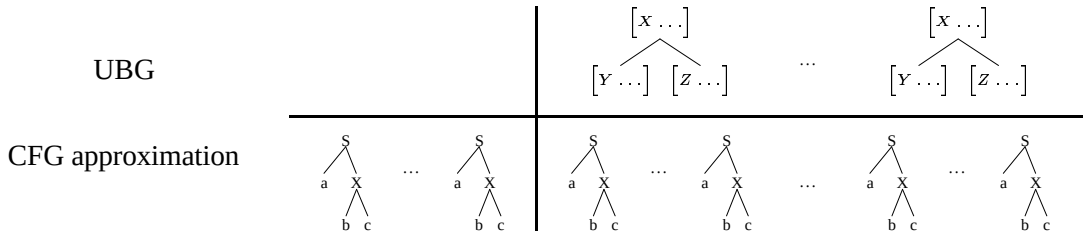

Figure 1: The readings of a sentence, analyzed by a UBG (top) and its CFG approximation (bottom). The picture illustrates that (i) each UBG reading of the sentence is associated with a non-empty set of syntax trees according to the CFG approximation, and (ii) that the sentence may have CFG trees, which can not be replayed by the UBG, since the CFG overgenerates (or at best is a correct approximation of the UBG).

1994) or PATR-II (Shieber, 1985) into context-free grammars (CFG). The method was introduced by Kiefer and Krieger (2000).

The approximation method can be seen as the construction of the least fixpoint of a certain monotonic function and shares similarities with the instantiation of rules in a bottom-up passive chart parser or with partial evaluation in logic programming. The basic idea of the approach is as follows. In a first step, one generalizes the set of all lexicon entries. The resulting structures form equivalence classes, since they abstract from word-specific information, such as FORM or STEM. The abstraction is specified by means of a restrictor (Shieber, 1985), the so-called *lexicon restrictor*. After that, the grammar rules are instantiated by unification, using the abstracted lexicon entries and resulting in derivation trees of depth 1. The *rule restrictor* is applied to each resulting feature structure (FS), removing all information contained only in the daughters of a rule. Additionally, the restriction gets rid of information that will either lead to infinite growth of the FSs or that does not constrain the search space. The restricted FSs (together with older ones) then serve as the basis for the next instantiation step. Again, this gives FSs encoding a derivation, and again the rule restrictor is applied. This process is iterated until a *fixpoint* is reached, meaning that further iteration steps will not add (or remove) new (or old) FSs to the set of computed FSs.

Given the FSs from the fixpoint, it is then easy to generate context-free productions, using the *complete* FSs as symbols of the CFG; see Kiefer and Krieger (2002). We note here that adding (and perhaps removing) FSs during the iteration can be achieved in different ways: either by employing feature structure equivalence $\equiv$ (structural equivalence) or by using FS subsumption $\sqsubseteq$. It is clear that the resulting CFGs will behave differently (see figure 4). An in-depth description of the method, containing lots of details, plus a mathematical underpinning is presented in (Kiefer and Krieger, 2000) and (Kiefer and Krieger, 2002). The application of the method to a mid-size UBG of English, and large-size HPSGs of English and Japanese is described in (Kiefer and Krieger, 2002) and (Kiefer et al., 2000).

3 A Novel Disambiguation for UBGs

(Kiefer and Krieger, 2000) suggest that, given a UBG, the approximated CFG can be used as a cheap filter during a two-stage parsing approach. The idea is to let the CFG explore the search space, whereas the UBG deterministically replays the derivations, proposed by the CFG. To be able to carry out the replay, during the creation of the CF grammar, each CF production is correlated with the UBG rules it was produced from.

The above mentioned two-stage parsing approach not only speeds up parsing (see figure 4), but can also be a starting point for an efficient stochastic parsing model, even though the UBG might encode an infinite number of categories. Given a training corpus, the idea is to move from the approximated CFG to a PCFG which predicts probabilities for the CFG trees. Clearly, the probabilities can be used for disambiguation, and more important, for ranking of CFG trees. The idea is, that the ranked parsing trees can be replayed one after another by the UBG (processing the most probable CFG trees first), establishing an order of best UBG parsing trees. Since the approximation always yields a CFG that is a superset of the UBG, it might be possible that derivation trees proposed by the PCFG can *not* be replayed by the UBG. Nevertheless, this behavior does not alter the ranking of reconstructed UBG parsing trees. Figure 1 gives an overview, displaying the readings

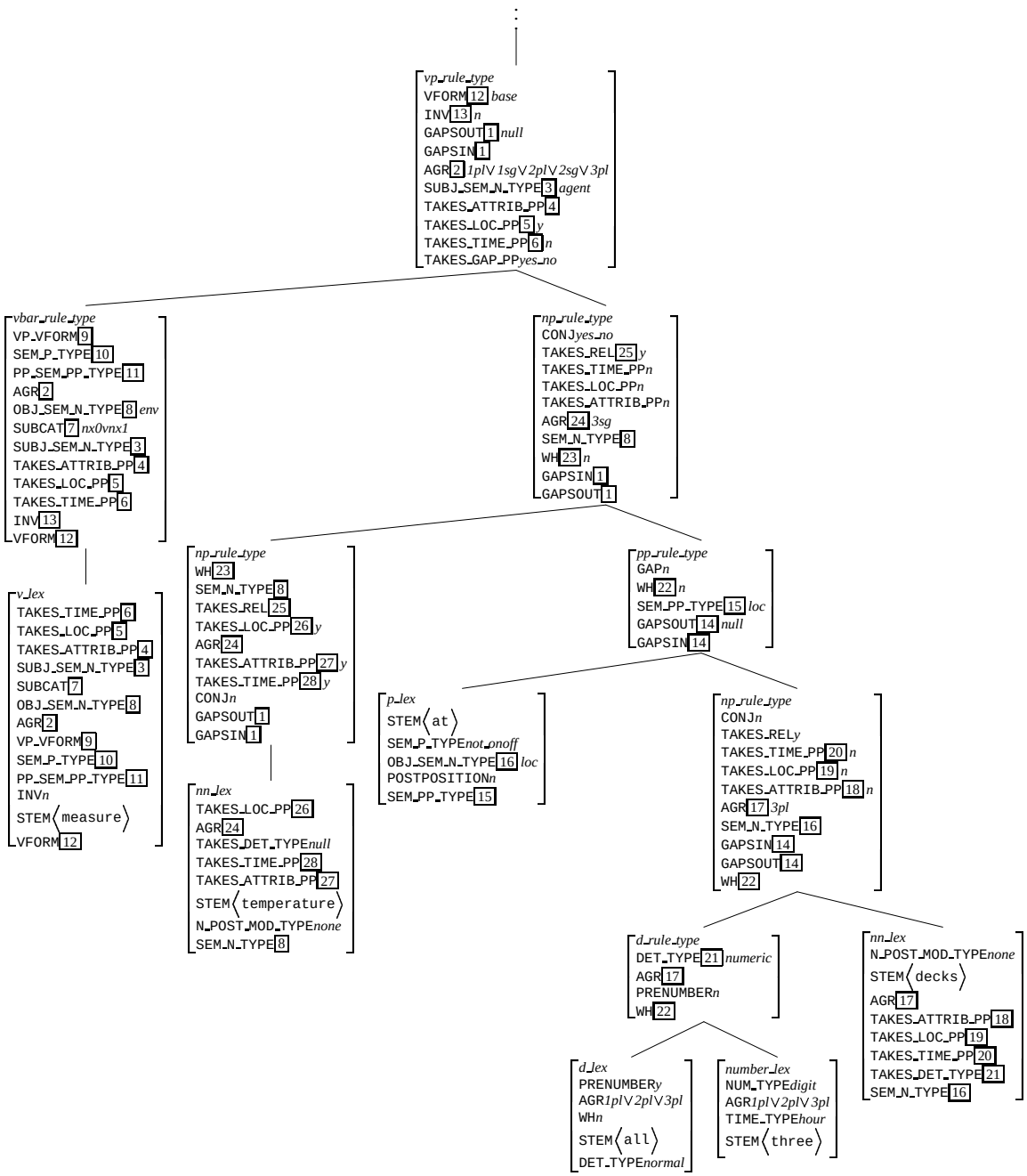

Figure 2: One of the two readings for the sentence *measure temperature at all three decks*, analyzed by the Gemini grammar. Note that the vertical dots at the top indicate an incomplete FS derivation tree. Furthermore, the FSs at the tree nodes are massively simplified.

of a sentence, analyzed by a UBG and its CFG approximation. Using this figure, it should be clear that a ranking of CFG trees induces a ranking of UBG readings, even if not all CFG trees have an associated UBG reading. We exemplify our idea in section 4, where we disambiguate a sentence with a *PP*-attachment ambiguity.

As a nice side effect, our proposed stochastic parsing model should usually *not* explore the full search space, nor should it statically estimate the parsing results afterwards, assuming we are interested in the most probable parse (or say, the two most probable results)—the disambiguation of UBG results is simply established by the dynamic ordering of most probable CFG trees during the first parsing stage.

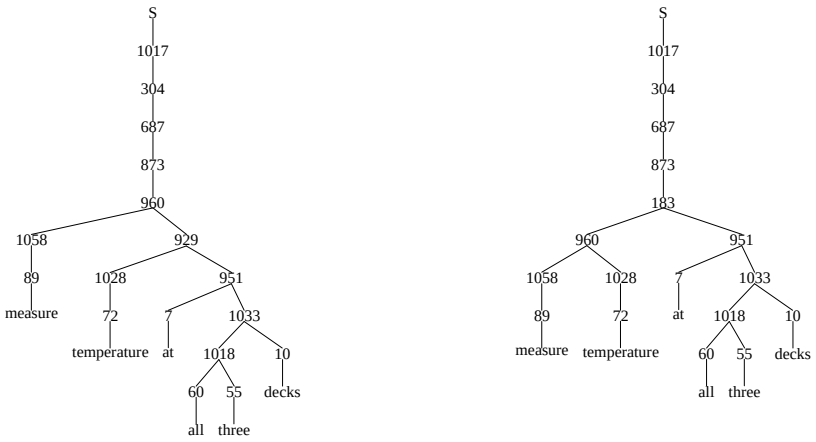

Figure 3: Alternative readings licensed by the context-free approximation of the Gemini grammar.

4 Experiments

Approximation. (Dowding et al., 2001) compared (Moore, 1999)'s approach to grammar approximation to (Kiefer and Krieger, 2000). As a basis for the comparison, they chose an English grammar written in the Gemini/CLE formalism. The motivation for this enterprise comes from the use of the resulting CFG as a context-free language model for the Nuance speech recognizer. John Dowding kindly provided the Gemini grammar and a corpus of 500 sentences, allowing us to measure the quality of our approximation method for a realistic mid-size grammar, both under $\equiv$ and $\sqsubseteq$ (see section 2).[1]

The Gemini grammar consisted of 57 unification rules and a small lexicon of 216 entries which expanded into 425 full forms. Since the grammar allows for atomic disjunctions (and makes heavy use of them), we ended in overall 1,886 type definitions in our system. Given the 500 sentences, the Gemini grammar licensed 720 readings. We only deleted the ARGS feature (the daughters) during the iteration and found that the original UBG encodes a context-free language, due to the fact that the iteration terminates under $\equiv$. This means that we have even obtained a *correct* approximation of the Gemini grammar. Table 4 presents the relevant numbers, both under $\equiv$ and $\sqsubseteq$, and shows that the ambiguity rate for $\sqsubseteq$ goes up only mildly.

We note, however, that these numbers differ from those presented in (Dowding et al., 2001). We could not find out why their implementation produces worse results than ours. They suggested that the use of $\sqsubseteq$ is the reason for the bad behaviour of the resulting grammar, but, as our figures show, this is not true, at least not for this grammar. Of course, using $\sqsubseteq$ instead of $\equiv$ can lead to substantially less restrictive grammars, but when dealing with complex grammars, there is—at the moment—no alternative to using $\sqsubseteq$ due to massive space and time requirements of the approximation process.

[1] A big *thank you* is due to Mark-Jan Nederhof who has written the Gemini-to-$\mathcal{TDL}$ converter and to John Dowding and Jason Baldridge for fruitful discussions.

	Gemini	$\equiv$	$\sqsubseteq$
# readings	720	720	747
ambiguity rate	1.44	1.44	1.494
#terminals	—	152	109
#nonterminals	—	3,158	998
#rules	57	24,101	5,269
#useful rules	57	19,618	4,842
running time (secs)	32.9	14.6	9.5
run time speed-up (%)	0	55.6	71.1

Figure 4: A comparison of the approximated CFGs derived under $\equiv$ and $\sqsubseteq$. The fixpoint for $\equiv$ ($\sqsubseteq$) was reached after 9 (8) iteration steps and took 5 minutes (34 seconds) to be computed, incl. post-processing time to compute the CF productions. The run time speed-up for two-stage parsing is given in the last row. The measurements were conducted on a 833 MHz Linux workstation.

Figure 2 displays one of the two readings for the sentence *measure temperature at all three decks*, analyzed by the Gemini grammar. The sentence is one of the 500 sentences provided by John Dowding. The vertical dots simply indicate that some less relevant nodes of the FS derivation tree have been omitted. The figure shows the reading, where the *PP at all three decks* is attached to the *NP temperature*. Due to space constraints, we do not show the second reading, where the *PP* is attached to the *VP measure temperature*.

Figure 3 shows the two syntax trees for the sentence, analyzed by the context-free approximation of the Gemini grammar, obtained by using $\sqsubseteq$. It

S	→	1017	(0.995)
1017	→	304	(0.472)
304	→	687	(0.980)
687	→	873	(1.000)
873	→	960	(0.542)
873	→	183	(0.330)
960	→	1058 929	(0.138)
960	→	1058 1028	(0.335)
183	→	960 951	(0.042)
1058	→	89	(1.000)
89	→	measure	(0.941)
929	→	1028 951	(0.938)
1028	→	72	(0.278)
72	→	temperature	(0.635)
951	→	7 1033	(0.286)
7	→	at	(0.963)
1033	→	1018 10	(0.706)
1018	→	60 55	(0.581)
60	→	all	(0.818)
55	→	three	(0.111)
10	→	decks	(1.000)

Figure 5: Fragment of the PCFG. The values in parenthesis are probabilities for grammar rules, gathered after two training iterations with the inside-outside algorithm.

is worth noting that both readings of the CFG approximation differ in *PP* attachment, in the same manner as the readings analyzed by the UBG itself. In the figure, all non-terminals are simply displayed as numbers, but each number represents a fairly complex feature structure, which is, in general, slightly *less* informative than an associated tree node of a possible FS derivation tree of the given Gemini grammar for two reasons. Firstly, the use of the $\sqsubseteq$ operation as a test generalizes information during the approximation process. In a more complex UBG grammar, the restrictors would have deleted even more information. Secondly, the flow of information in a local tree from the mother to the daughter node will not be reflected because the approximation process works strictly bottom up from the lexicon entries.

Training of the CFG approximation. A sample of sentences serves as input to the inside-outside algorithm, the standard algorithm for unsupervised training of PCFGs (Lari and Young, 1990). The given corpus of 500 sentences was divided into a training corpus (90%, i.e., 450 sentences) and a testing corpus (10%, i.e., 50 sentences). This standard procedure enables us (i) to apply the inside-outside algorithm to the training corpus, and (ii) to evaluate the resulting probabilistic context-free grammars on the testing corpus. We linguistically evaluated the maximum-probability parses of all sentences in the testing corpus (see section 5). For unsupervised training and parsing, we used the implementation of Schmid (1999).

Figure 5 shows a fragment of the probabilistic context-free approximation. The probabilities of the grammar rules are extracted after several training iterations with the inside-outside algorithm using the training corpus of 450 sentences.

Disambiguation using maximum-probability parses. In contrast to most approaches to stochastic modeling of UBGs, PCFGs can be very easily used to assign probabilities to the readings of a given sentence: the probability of a syntax tree (the reading) is the product of the probabilities of all context-free rules occurring in the tree.

For example, the two readings of the sentence *measure temperature at all three decks*, as displayed in figure 3, have the following probabilities: $2.25 \cdot 10^{-12}$ (first reading on the left-hand side) and $1.49 \cdot 10^{-13}$ (second reading on the right-hand side). The maximum-probability parse is therefore the syntax-tree on the left-hand side of figure 3, which is the reading, where the *PP at all three decks* is attached to the *NP temperature*.

A closer look on the PCFG fragment shows that the main contribution to this result comes from the two rules 929 →1028 951 (0.938) and 183 →960 951 (0.042). Here, the probabilities encode the statistical finding that *PP-to-NP* attachments can be expected more frequently than *PP-to-VP* attachments, if the context-free approximation of the Gemini grammar is used to analyze the given corpus of 500 sentences.

5 Evaluation

Evaluation task. To evaluate our models, we used the testing corpus mentioned in section 4. In a next step, the correct parse was indicated by a human disambiguator, according to the intended reading. The average ambiguity of this corpus is about 1.4 parses per sentence, for sentences with about 5.8 words on average.

Our statistical disambiguation method was tested on an *exact match* task, where exact correspondence of the manually annotated correct parse and the most probable parse is checked. Performance on this evaluation task was assessed according to the following evaluation measure:

$$\text{precision} = \frac{\#\text{correct}}{\#\text{correct}+\#\text{incorrect}}$$

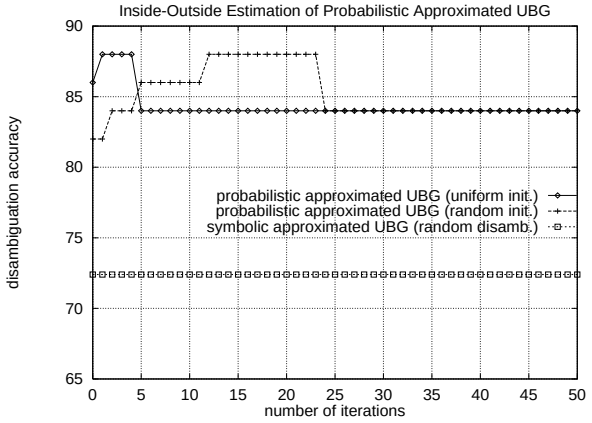

Figure 6: Precision on exact match task in number of training iterations for probabilistic context-free approximations, starting with uniform and random probabilities for the grammar rules. Baseline is the disambiguation accuracy of the symbolic approximated UBG.

where "correct" and "incorrect" specifies a success or failure on the evaluation tasks, resp.

Evaluation results. First, we calculated a random baseline by randomly selecting a parse for each sentence of the test corpus. This baseline measures the disambiguation power of the pure symbolic parser and was around 72% precision.

Optimal iteration numbers were decided by repeated evaluation of the models at every iteration step. Fig. 6 shows the precision of the models on the exact match task plotted against the number of iterations of the training algorithm. The baseline represents the disambiguation accuracy of the symbolic approximated UBG which is clearly outperformed by inside-outside estimation, starting with uniform or random probabilities for the rules of the CF approximation. A clear overtraining effect occurs for both cases (see iterations ≥ 5 and ≥ 24, resp.).

A comparison of the models with our random baseline shows an increase in precision of about 16%. Although we tried hard to improve this gain by varying the starting parameters, we wish to report that we found no better starting parameters than uniform probabilities for the grammar rules.

6 Related Work and Discussion

The most direct points of comparison of our method are the approaches of Johnson et al. (1999) and Riezler et al. (2000), esp. since they use the same evaluation criteria than we use.

In the first approach, log-linear models for LFG grammars were trained on treebanks of about 400 sentences. Precision was evaluated for an ambiguity rate of 10 (using cross-validation), and achieved 59%. If compared to this, our best models achieve a gain of about 28%. However, a comparison is difficult, since the disambiguation task is more easy for our models, due to the low ambiguity rate of our testing corpus. However, in contrast to our approach, supervised training was used by Johnson et al. (1999).

In the second approach, log-linear models of LFG grammars were trained on a text corpus of about 36,000 sentences. Precision was evaluated on 550 sentences with an ambiguity rate of 5.4, and achieved 86%. Again, a comparison is difficult. The best models of Riezler et al. (2000) achieved a precision, which is only slightly lower than ours. However, their results were yielded using a corpus, which is about 80 times as big as ours.

Similarly, a comparison is difficult for most other state-of-the-art PCFG-based statistical parsers, since different training and test data, and most importantly, different evaluation criteria were used.

7 Conclusion

This paper concerns the problem of how to disambiguate the readings of sentences, analyzed by a given UBG.

We presented a novel approach to disambiguation for UBGs, shifting the responsibility to simpler CF models, obtained by the approximation of the UBG.

In contrast to earlier approaches to disambiguation for UBGs, our approach can be effectively applied in practice, enables unsupervised training on free text corpora, as well as efficient disambiguation, and is mathematically clean.

We showed that our novel approach is feasible for a mid-size UBG of English. Evaluation of an unsupervised trained model achieved a precision of 88% on an exact match task.

8 Acknowledgements

This research was supported by the German Federal Ministry for Education, Science, Research, and Technology under grant no. 01 IW 002 and EU grant no. IST-1999-11438.

References

Steven Abney. 1997. Stochastic attribute-value grammars. *Computational Linguistics*, 23(4):597–618.

Rens Bod and Ron Kaplan. 1998. A probabilistic corpus-driven model for lexical-functional analysis. In *Proceedings of COLING/ACL-98*.

Taylor L. Booth and Richard A. Thompson. 1973. Applying probability measures to abstract languages. *IEEE Transactions on Computers*, C-22(5):442–450.

Gosse Bouma, Gertjan van Noord, and Robert Malouf. 2001. Alpino: Wide-coverage computational analysis of dutch. In *Computational Linguistics in The Netherlands 2000*.

Chris Brew. 1995. Stochastic HPSG. In *Proceedings of the EACL-95*, Dublin.

Ted Briscoe and John Carroll. 1993. Generalized probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19(1):25–59.

John Dowding, Beth Ann Hockey, Jean Mark Gawron, and Christopher Culy. 2001. Practical issues in compiling typed unification grammars for speech recognition. In *Proceedings of ACL-2001*, pp. 164–171.

Andreas Eisele. 1994. Towards probabilistic extensions of constraint-based grammars. In Jochen Dörre, editor, *Computational Aspects of Constraint-Based Linguistic Description II*, pp. 3–21. DYANA-2 Deliverable R1.2.B.

Joshua Goodman. 1997. Probabilistic feature grammars. In *Proceedings of the International Workshop on Parsing Technologies*.

Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proceedings of ACL-99*.

Bernd Kiefer and Hans-Ulrich Krieger. 2000. A context-free approximation of Head-Driven Phrase Structure Grammar. In *Proceedings of the 6th International Workshop on Parsing Technologies, IWPT2000*, pp. 135–146.

Bernd Kiefer and Hans-Ulrich Krieger. 2002. A context-free approximation of Head-Driven Phrase Structure Grammar. In *Efficiency in Unification-Based Processing*. CSLI Lecture Notes.

Bernd Kiefer, Hans-Ulrich Krieger, and Melanie Siegel. 2000. An HPSG-to-CFG approximation of Japanese. In *Proceedings of COLING-2000*, pp. 1046–1050.

K. Lari and S. J. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56.

Robert C. Moore. 1999. Using natural-language knowledge sources in speech recognition. In Keith Ponting, editor, *Computational Models of Speech Pattern Processing*. Springer.

Miles Osborne. 2000. Estimation of stochastic attribute-value grammars using an informative sample. In *Proceedings of COLING-2000*.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. University of Chicago Press.

S. Riezler, D. Prescher, J. Kuhn, and M. Johnson. 2000. Lexicalized stochastic modeling of constraint-based grammars using log-linear measures and EM training. In *Proc. of ACL-2000*.

Helmut Schmid, 1999. *LoPar. Design and Implementation*. Insitut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Helmut Schmid. 2002. Probability models for unification-based grammars. Internal report, IMS, University of Stuttgart.

Stuart M. Shieber. 1985. Using restriction to extend parsing algorithms for complex-feature-based formalisms. In *Proceedings of ACL-85*, pp. 145–152.